Development of a Freeze-out Surface for 3+1 Dimensional Hydrodynamics

Maciej Andrzej Stankiewicz

November 2006

Thesis submitted in fulfilment of
the requirements for the degree of Master of Science
at the University of Cape Town

Abstract

A review of a macroscopic description of heavy ion collisions is presented, with focus on hydrodynamics. The various models are discussed and motivation is provided for using full 3+1 dimensional hydrodynamics. Numerical freeze-out is then discussed, and a numerical development of the freeze-out hypersurface is presented. Possible algorithmic extensions are described.

Contents

List of Figures

Chapter 1

Introduction

The twentieth century has been a successful one for the field of physics. In the early stages it witnessed the development of quantum mechanics and special relativity. Soon afterwards general relativity (GR) was introduced, together with its implication that observed gravitational interactions can be explained as geodesical motions in a curved space-time. This conceptual leap in understanding had a profound impact on the physics community. Around the midpoint stage there was another major development - this time on the side of field theory. The theory of Quantum ElectroDynamics (QED) was developed to explain electromagnetic interactions and it, in turn, implied that in any system all possible interactions resulting in an observed final state do, in fact, happen. Unrealistic as they appear, both GR and QED have been tested and found to be accurate to an astonishing degree, and have since been accepted as excellent theories.

After the stunning success of QED, the physics community tried to apply the field theory approach to other interactions. To find a complete description of the strong interaction became the ultimate goal. The domain of field theory has been further developed, and what we believe to be the fundamental Lagrangian of the strong interaction was discovered a little over 50 years ago [6]. In contrast to QED, the theory of strong interactions contains three 'colour' charges (hence QCD – Quantum ChromoDynamics) and lives in SU(3) colour space (as opposed to QED which lives in U(1) space). This implies that the theory must be insensitive to 'rotations' in SU(3). However, these rotations are non-Abelian, and this results in the force-mediating gluons not being colour-neutral (the gluons carry a dual colour-anticolour charge), which allows them to self-interact. In addition, it turns out that there are 8 distinct gluons (corresponding to the 8 linearly independent traceless 3×3

Hermitean matrices which are the generators of SU(3)). It is these extra intrinsic complications in the theory of QCD that are responsible for nuclear forces being so strong and short-ranged. (In the simpler theory of QED there is only one charge and the photons, being charge-neutral, do not, to first order, self-interact.)

As the analytic basis of QCD is now well known and established, in principle we ought to know everything about the strong interaction. However, there are, as yet, no known mathematical techniques to find closed form solutions to any desired observables. Although there is a large interest in developing that field, the immediate future is rather bleak and most people involved have moved onto other ways of extracting information from the theory.

One intriguing feature of QCD is that the coupling constant α_s decreases with an increase in momentum transfer. This leads to two phenomena which have been termed 'asymptotic freedom' and 'quark confinement'. These radically different behaviours make it difficult to construct a single, coherent, qualitative description of the systems in question. They do, however, leave room for partial descriptions of various aspects of the theory, using some general assumptions. Perturbation and lattice theory are two such examples, and both have produced results in their respective domains.

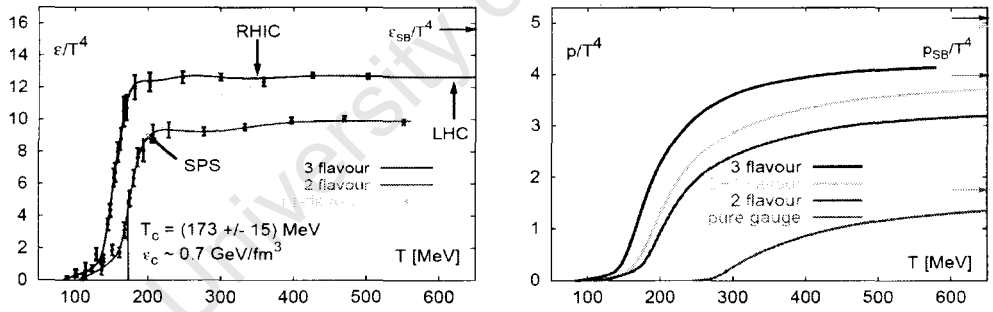

Figure 1.1: The normalized energy density ε/T^4 and pressure p/T^4 from lattice QCD [14] for 2 and 3 light quark flavours, as well as for 2 light +1 heavier (strange) quark flavour. Horizontal arrows on the right indicate the corresponding Stefan-Boltzmann values for a non-interacting quark-gluon gas.

A prediction of lattice QCD [14] (which works with zero net baryon density) is that, at a certain critical temperature and energy density (see figure 1.1 above), the quarks and gluons, which remain bound up inside hadrons and then inside the nuclei that surround us, will break their hadronic bags and form a different state of matter. This phase transition is predicted to occur

at $T_{crit} \approx 175\text{MeV} \sim 2 \cdot 10^{12}\text{K}$. At this temperature, it is predicted that the coloured quarks and gluons will form a state with many more partonic degrees of freedom, and that they will become essentially freely roaming particles within their medium. As such, the partons themselves will become the constituents that determine the thermodynamic properties of the system. This hypothesized state has been termed the "Quark-Gluon Plasma". A QGP is thought to have existed in the very early stages of the universe, shortly after the big bang, but due to expansion the universe would have cooled to below the critical temperature and the quarks became confined inside hadrons and then nuclear matter which we see around us today.

Interestingly enough, this is not the only new phase of matter that has been predicted by QCD. Recent work has shown strong evidence to suggest that at very high net quark densities (more than $10\rho_0$, with $\rho_0 \sim 3 \times 10^{14}\text{g/cm}^3$, the nuclear density) and relatively low temperatures ($T < 50\text{MeV}$), quark matter may undergo a different phase transition [15]. In this system the quarks are expected to pair up, creating coloured bosonic pairs. These pairs would then form the constituents of the system and (at sufficiently low temperatures) create a bosonic condensate. This possible new state has been termed the "Colour SuperConductor" or CSC.

It has become the particle physicist's goal to recreate at least one of these new states of matter in a laboratory and investigate its properties. We have, as yet, no means of creating a CSC (it might exist inside neutron stars). It's thought, however, that one can recreate a QGP by creating a system with a very high temperature (and moderately high quark density). Heavy ion collisions at ultra-relativistic energies do just that.

1.1 Heavy ion collisions as a probe for a new state of matter

Since the 1960's, experiments have been set up to collide nuclei with other sub-atomic particles (electrons, protons, other nuclei). Originally they were collided at what is now considered to be very low energies (as well as being fixed target experiments), and were used to observe and study basic particle physics and particle production. However, as accelerator physics was refined, the collision energies were increased and experiments are now created in a colliding beam setup. This has the advantage of creating a much higher centre-of-mass energy and also makes the centre-of-mass frame the lab

frame, provided the two sets of projectiles are identical, which is excellent experimentally, as there is no inherent boost to all the particles involved.

As the energies afforded to the nuclei increased, a point was reached where the nuclear collisions became sufficiently energetic that the original nuclei appeared to become 'melted' during the initial collision, and after a short time they formed an almost continuous state of strongly interacting matter, the constituents of which were not nucleons. This has been studied [2, 3, 4], and it has been accepted by most people that this non-standard form of matter is a strongly-interacting 'gas' consisting of base hadronic matter (the 'gas particles' are simply the hadrons, including the short-lived resonances). It is a 'gas' in the sense that its constituent 'particles' have short mean-free paths before they collide with another particle, and these collisions are originally inelastic. Such a system will naturally expand and cool, with the hadronic 'gas particles' getting further apart and losing energy until no further particle production can take place (freeze-out) and the particles fly away to the detectors. There is strong evidence to support this model, to be discussed in more detail in the next chapter.

A major difficulty is to discover what happens in the system initially. In the moments (order $\sim$ 1fm/c) immediately after a collision, at RHIC energies, the energy density and temperature of the system are significantly higher than the critical values predicted by lattice QCD, allowing for a possible phase transition to occur. This we cannot observe, as all the hadrons are produced in the next stage of collision evolution. It is a similar proposition to that of investigating the early universe before the matter-energy decoupling.

1.2 QGP signatures

Due to the component quarks and gluons of a QGP being confined, it will never be possible to observe a QGP directly. For heavy ion collisions, its existence has to be inferred from the observed distributions of hadrons, leptons and photons. This is a highly non-trivial task as the hadrons reflect the conditions at freeze-out, by which time the memory of the original partonic state may have been completely erased. The directly emitted leptons and photons suffer far fewer interactions before detection and are a better probe of the early conditions of the system. They are, however, far less numerous than the hadrons, and there is some difficulty extracting them from the indirect leptons and photons produced by hadronic decays.

There have been other propositions for inferring QGP formation. J/ψ suppression [17], as well as strangeness enhancement [2, 16] are the better researched topics. The focus on heavy, flavoured quarks is common as the creation of the heavier quarks is expected to be easier in the quark-gluon plasma (and the J/ψ suppression comes from the charm anti-charm quarks not bonding into a hadron automatically after creation, but rather living free in the QGP and hadronizing with other, light quarks).

An alternative way of inferring the phase transition is to consider the hydrodynamical flow of the system. A hadronic gas, a QGP and any possible mixed phase that might occur will have different pressure gradients and speeds of sound, which will yield different flow evolutions and final particle spectra.

One of the major remaining questions is 'How will a QGP behave?', and this is one that does not, as yet, have a definite answer. The theorists may try to explain the observed phenomena using known hadronic physics, but, if they are unsuccessful, this does not automatically force a phase transition to have occurred. Recently the experiments at RHIC have announced that they created "a perfect fluid" [28], and this has been attributed to a hydrodynamical flow of the QGP [29]. However, nobody can, as yet, convincingly *prove* that this fluid is the elusive QGP. This issue will not be fully laid to rest until a definite QGP effect is predicted, and only *later* found to be true.

1.3 Objectives and structure of thesis

In the next chapter, the goal is to present arguments which describe the collective models of heavy ion collisions, as well as some overlaps with other techniques. An outline of the ideal hadron gas model is presented, together with the method for obtaining particle spectra. The main focus though is on hydrodynamics: the well studied dimensionally reduced models of Landau (complete stopping) and Bjørken (nuclear transparency) are briefly discussed. The Bjørken formalism, which is often used in hydrodynamics, is also presented. The shortcomings of these simplified models in the description of heavy ion collisions will be demonstrated, and motivation for using full 3+1 dimensional hydrodynamics will be presented.

Chapter 3 will refer entirely to the freeze-out: the mechanism and criterion, and finally a full description of the new approach to the numerical implementation of a freeze-out hypersurface, with an outline of spectrum construction. Possible algorithmic extensions are also presented. This is the main objective of this thesis.

Chapter 2

Macroscopic Description of Heavy Ion Collisions

There are many active approaches to describing heavy ion collisions. Most are numerical in essence and involve major development teams working on their models in an attempt to find a numerical way of describing the collisions as accurately as possible. Widely used are programs such as HIJING (Heavy Ion Jet INteraction Generator) [30] and UrQMD (Ultra-relativistic Quantum Molecular Dynamics) [31]. These are both microscopic transport models, though they differ significantly in the details. HIJING uses perturbative QCD reactions, as well as the Glauber model for describing A+A collisions. It also implements string transport to model the soft processes in the later stages of a collision. UrQMD, on the other hand, is a hadronic transport model, which tracks the evolution of the system by using measured and extrapolated hadronic cross sections.

The focus in this thesis is not on these models, which track the collision evolution on a particle level, but rather on hydrodynamics, where the system is modelled as a continuous fluid and the constituent partons have no exterior presence. The system's evolution is governed by its macroscopic parameters such as energy density, temperature and pressure.

After a certain time in hydrodynamic expansion, the system will become too dilute to be modelled as a continuous fluid. Furthermore, at a certain temperature, the system will become too cool for inelastic hadronic reactions to take place, and the particles that exist in the ensemble will remain that way until they naturally decay. This condition at which inelastic collisions cease and the particle multiplicities become fixed (frozen) has been termed

the "freeze-out criterion". It is at this freeze-out that the hydrodynamical fluid is transformed into particles, the spectra of which can then be compared to experiment.

2.1 Collision mechanism

Consider the collision of two nuclei at relativistic speeds. Although one can collide different nuclei (such as d+Au at RHIC 2002), in order to create a 'large' system in which a QGP might be created both target and projectile have to be heavy ions. Hence it is now common to use identical nuclei. For colliding beam experiments, this will also give both nuclei equal energies, and the centre of mass will be stationary relative to the lab frame. The tags 'target' and 'projectile' lose their meaning, but still remain in use in some cases as a convenient way of distinguishing between the two nuclei.

2.1.1 Geometry

Firstly, let us recap the co-ordinate system which is commonly used to describe nuclear collisions. The centre of momentum of the two nuclei is defined as the origin of momentum space, and the collision point sets the origin along the beam direction. By convention, the nuclei travel along the z-axis. The plane orthogonal to the beam direction ($z = 0$) is called the *transverse plane*.

The nuclei are Lorentz contracted in the z-direction and are assumed to be circular in the transverse (x, y) plane. The level of overlap of the two nuclei is referred to as *centrality*. As the nuclei are dispersed across a macroscopically small, but nonetheless finitely tight beam (~ 1 mm^2), a collision between two nuclei will never be exactly head-on. At the moment of impact, we define the *impact parameter* $\vec{b}$ of the collision as the vector distance between the projections of the centres of the two nuclei onto the transverse plane. The x-direction is then defined as the direction of this impact parameter $\vec{b}$, and the y-direction is defined orthogonally, while maintaining a right-handed co-ordinate system. The plane $y = 0$ is known as the *reaction plane*, and is spanned by $\vec{b}$ and $\vec{z}$.

The centrality of a collision is given in terms of the impact parameter $b = |\vec{b}|$, and is always quoted in units of fm. Centrality is an effective measure of the size of the system, as it is monotonically related to the number of participating nuclei in the collision.

2.1.2 Glauber model

In the center of mass frame, each nucleus is travelling at a relativistic speed and gets Lorentz contracted by a factor of γ. For all RHIC collisions (energy ranges from $56A$ GeV to $200A$ GeV, hence $30 \leq \gamma \leq 106$) the nuclei get contracted to nuclear 'pancakes'. The nucleons which participate in a collision are (to a good approximation) the ones which geometrically overlap the other nucleus in the transverse plane. Formally, one uses the Glauber approximation [18][1].

Before the collision, the density of nucleons in a nucleus of mass number A is given by the Woods-Saxon density function

$$\rho_A(r) = \frac{\rho_0}{e^{(r-R_A)/\xi} + 1}, \tag{2.1}$$

where R_A is the classical radius, ξ is the surface diffuseness and ρ_0 is a nuclear density parameter, which is obtained from the normalization

$$\int \rho_A(r) d^3r = A. \tag{2.2}$$

For RHIC (Au+Au) collisions, $A = 197$, and one finds that $R_A = 6.37$ fm, $\xi = 0.54$ fm, and hence $\rho_0 = 0.17$ fm^{-3}. The LHC will be using lead-lead collisions, so the numbers will change slightly.

Now, recalling that the nuclei are highly contracted, it makes sense to define the *nuclear thickness function*, which is obtained by integrating the nuclear density over the beam (contraction) direction:

$$T(x, y) = \int_{-\infty}^{+\infty} \rho_A(x, y, z) dz. \tag{2.3}$$

This gives a function of x and y, which, though defined in the rest frame of the nucleus, refers to the number of nucleons associated with a point in the transverse plane. Hence it is independent of boosts in the beam direction, and thus remains the same in the lab frame.

This nuclear thickness function gives the nuclear densities of the colliding nuclear pancakes, which are used to initialize the system. It can also be used to accurately predict the number of nucleons which will be involved in a collision, as well the number of initial nucleon-nucleon interactions.

[1] This model is attributed to the 2005 nobel laureate Roy Glauber. Although it is widely used in heavy ion physics, Prof. Glauber, who works primarily in the field of theoretical optics, was until recently unaware of the model's application in this field.

Participant nucleons

The nucleons which interact at least once in a collision are called participant nucleons. For lower energy experiments (such as those at the SPS) it was observed that the number of particles emitted is linearly related to the number of participant nucleons [21]. Deviations from this scaling have since been observed at RHIC energies [23, 25]. The number of participants is still expected to be responsible for the bulk of soft particle production, as the energy afforded to the system scales linearly with the number of participants.

Statistical considerations can be used to establish an algebraic form for the density of participant nucleons in the transverse plane in terms of the nuclear thickness functions of the two colliding nuclei [1, 22]. For nuclei of mass numbers A and B, this is given by

$$\begin{aligned}\langle N_{Part}(b)\rangle &= A\iint T_A(x,y)\left[1-\left(\frac{1-\sigma_{in}T_B(x-b,y)}{B}\right)^B\right]dx\,dy \\ &\quad +B\iint T_B(x-b,y)\left[1-\left(\frac{1-\sigma_{in}T_A(x,y)}{A}\right)^A\right]dx\,dy, \quad (2.4)\end{aligned}$$

where T_A and T_B are the nuclear thickness functions of the two nuclei and σ_{in} is the inelastic nucleon-nucleon cross section. At RHIC energies, $\sigma_{in} \approx 40$mb. For symmetric collisions, the two terms become identical, and:

$$\langle N_{Part}(b)\rangle = 2A\iint T_A(x,y)\left[1-\left(\frac{1-\sigma_{in}T_A(x-b,y)}{A}\right)^A\right]dx\,dy. \quad (2.5)$$

An evaluation of $\langle N_{Part}(b)\rangle$ shows a monotonic dependence on the impact parameter b (see figure 2.1). By observing and counting the number of non-participant nucleons in a collision[2], the centrality of each collision can be determined. The observed particle multiplicities and distributions are thus matched up with the appropriate initial conditions (nuclear overlap). With this, the study of specific classes (fully central, very peripheral, etc) of collisions can be done.

One intriguing aspect of this formalism is that even in a fully central collision ($b = 0$) the number of participant nucleons is not the total number of nucleons of the two nuclei. For RHIC (Au+Au) collisions, the predicted number is ≈ 380, mainly due to the 'fuzzy' edges of the gold nuclei not necessarily interacting with each other.

[2]RHIC uses Zero Degree Calorimeters - scintillators placed along the z-axis tuned to detect the neutrons from fragmentation of the colliding nuclei [33].

Binary collisions

Another interesting quantity to obtain is the number of hard nucleon-nucleon interactions (binary collisions). Hard collisions are those where the constituent quarks interact with an extremely high momentum transfer - this results in a different class of particle production: the quarks that collided fly away from each other with very high momenta, in the process creating a mass of quark-antiquark pairs due to string fragmentation. These quarks pair up to make 'jets' of very high momentum particles.

The standard way of counting the number of binary collisions is by taking each nucleon in one nucleus, and counting the number of nucleons in the target projectile that it would interact with if it travels straight through. As such, the average number of binary collisions is:

$$\langle N_{BC}(b) \rangle = \iint \sigma_{in} T(x,y) T(x-b,y)\, dxdy. \tag{2.6}$$

However, far less jets are observed than predicted. It is expected that jets get absorbed into the medium if they pass through a dense region of the collision. As the absorption rate will depend on the medium, jet quenching is being studied as a possible QGP effect [19].

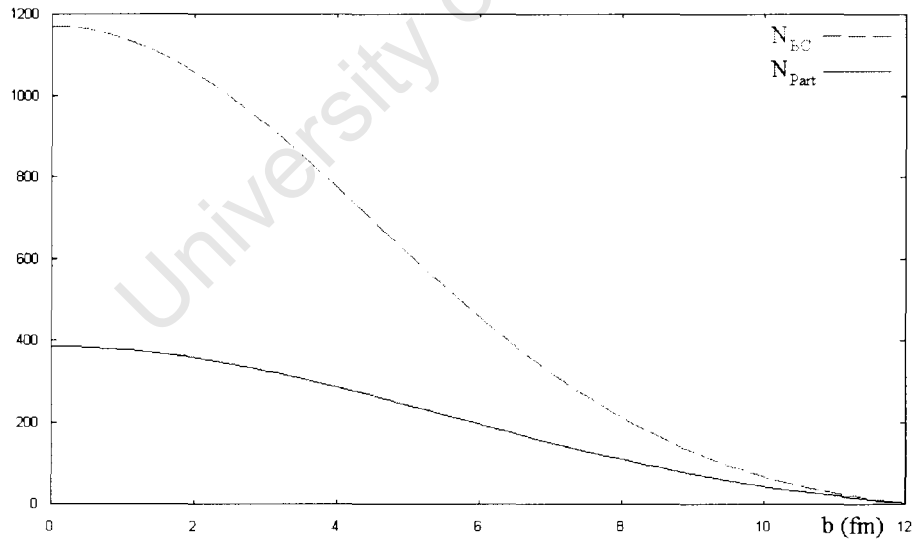

Figure 2.1: Centrality dependence of the total number of participants and binary collisions for Au+Au collisions, with $\sigma_{in} \approx 40$mb [1].

2.2 Rapidity and pseudorapidity

In describing particle production in heavy ion collisions, theoreticians like using the longitudinal rapidity y due to its Lorentz invariance and the workable form into which the statistical integrals translate. Spectra are usually given in the form $\frac{dN}{dy}$ – binned in rapidity intervals. However, rapidity is a theoretical construct and, although it is well defined for every particle, to determine the rapidity experimentally, one requires knowledge of the particle's total energy and longitudinal momentum. Specifically:

$$y \equiv \frac{1}{2} \ln \left(\frac{E + p_z}{E - p_z} \right) = \tanh^{-1} \left(\frac{p_z}{E} \right). \tag{2.7}$$

Obtaining these parameters is a very non-trivial process. The longitudinal momentum uniquely defines the transverse momenta, if the particle is assumed to be moving in a straight line before the first impact with a detector. To find the energy and momentum, one is effectively finding the mass of the particle in question. There are several techniques currently used for particle identification (PID).

2.2.1 Measuring rapidity

Some detectors, such as the PHOBOS spectrometer, use many layers of silicon wafer, which record how much energy is deposited when a charged particle passes through them. Applying a strong homogenous magnetic field orthogonal to the spectrometer, the charged particles curve between each layer of wafer, and the curvature increases as the particles continue losing energy. By analyzing the curvatures for each section (and knowing the differences in energies between them), it is possible to reconstruct the original energy of the particle and, after that, the particle's charge-mass ratio, and hence its mass[3]. An important aspect here is that very large numbers of particles are produced in each collision and, in order to reconstruct every particle 'track', the impact points on the individual silicon wafers have to be matched up correctly - a highly non-trivial task in its own right.

There are more advanced detectors available - such as the time projection chamber (TPC) in the STAR experiment at RHIC (and in the future ALICE

[3]Virtually all charged particles have a charge of $\pm e$. The exceptions are the Δ^{++}, Σ_c^{++} and some of the hypothesized pentaquarks. All these particles decay via the strong interaction, so will not be detected directly by the detectors.

experiment at LHC). These contain a cylindrical chamber filled with gas. The particles emitted in the collision ionize the gas and bend in the magnetic field (this time in the longitudinal direction), and, due to a 'small' electric field, the ionized electrons drift towards the edge of the chamber where their arrival times are recorded. By timing the arrival of the electrons, the time-position of their ionization can be accurately determined. Hence one finds a curved track for every charged particle that passed through the TPC, and again it can be PID'd and its energy determined.

There are several problems with these methods. They are both extremely computation intensive and are very susceptible to pileup (too many particles). Furthermore, in the case of the TPC, the gas in the chamber needs to be flushed on a regular basis, as any accumulated charge will render the data useless (it will effect the paths of the ionized electrons and make track reconstruction impossible), and this also limits the possible size of the TPC.

The two methods outlined above also rely on the particles bending sufficiently in the magnetic field for the curvature (and changes in curvature) to be accurately measured. For particles with very high transverse momenta, this becomes impossible. There are other approaches to identifying particles as well - the time of flight (TOF) detectors in PHOBOS are very high time-resolution crystals which, when placed 'far' from the collision point, can be used to measure the speed of the particles. This method clearly cannot be used effectively everywhere as $v \rightarrow c$ for high energies, but is an excellent detector at mid-rapidity ($y = 0 = p_z$), where all the momentum is in the transverse direction and is considerably smaller than most longitudinal momenta, as it had to be completely generated within the collision. By combining the data of energy loss in silicon with the speeds generated by the TOF walls, it is possible to identify particles with p_T up to 1-2 GeV (depending on species) [33].

2.2.2 Experimental approach

When the designs for experiments are drawn up, they are based primarily on the following principles:

- Measure what the theorists are predicting should happen in order to verify or dismiss current models.
- Measure everything else that one can, constrained by the budget, without hindering any other experimental elements.

To be consistent with the above, RHIC was designed with four very different experimental setups. However, each detector has a fixed geometrical setup and, as such, cannot be used to take data in terms of a dynamic parameter like rapidity. Rather, experimentalists use *pseudorapidity* instead, which is defined as

$$\eta = \frac{1}{2}\log\left(\frac{p+p_z}{p-p_z}\right) = \tanh^{-1}\left(\frac{p_z}{p}\right). \tag{2.8}$$

Now observe that

$$\tanh\eta = \frac{p_z}{p} = \frac{p_z}{\sqrt{p_z^2+p_T^2}}, \tag{2.9}$$

which transforms into

$$\frac{p_T}{p_z} = \sqrt{\frac{1-\tanh^2\eta}{\tanh^2\eta}} = \sqrt{\frac{\cosh^2\eta-\sinh^2\eta}{\sinh^2\eta}} = \frac{1}{\sinh\eta}. \tag{2.10}$$

This establishes a very clear purely geometrical representation of the pseudorapidity. In terms of the opening angle θ from the beam direction

$$\eta = \mathrm{Asinh}\left(\frac{p_z}{p_T}\right) = \mathrm{Asinh}\left(\frac{1}{\tan\theta}\right). \tag{2.11}$$

Many experiments use a 2π coverage in ϕ. This is useful in that there is no bias towards any transverse direction for a given opening angle, and is essential for measuring anisotropies in ϕ, which arise for all peripheral collisions. To establish such a coverage, there is usually some cylinder placed around the beam-pipe at mid-rapidity (such as the STAR TPC or the PHOBOS octagon), and coverage at forward angles can be implemented likewise (STAR's forward TPCs) or using other coverage methods (the PHOBOS rings are disks of silicon wafer). All these components are specified as having coverage in *pseudorapidity* η.

Experimentalists treat η as the important, observable parameter in particle production. Even the terms 'mid-rapidity' and 'forward rapidity' refer to pseudorapidity. For colliding beam experiments, it is clearly impossible to have a full 4π detection coverage, although one can cover virtually all space apart from the beam-pipes. The PHOBOS detector is designed in exactly such a way and has a silicon coverage (which detects charged particles only) of $-5.4 < \eta < 5.4$. Translated into angles, this means that every particle which is emitted at more that $0.5°$ from the beam direction will be observed [33]. The drawback is that PHOBOS is little more than a multiplicity counter in each η bin. For comparison, STAR's TPC only has a coverage of $-0.9 < \eta < 0.9$, which corresponds to an opening angle of $\theta \approx 44°$. It can, however, track and identify virtually all charged particles with momenta above 100MeV [34].

2.2.3 y - η correspondence

The implicit similarity in the definitions of y and η suggests a direct similarity between the two parameters. One can find a relationship in terms of the Lorentz γ factor:

$$\tanh y = \frac{p_z}{E} = \frac{p}{E}\frac{p_z}{p} = \frac{\sqrt{E^2 - m_0^2}}{E}\tanh\eta = \sqrt{1-\gamma^{-2}}\tanh\eta. \tag{2.12}$$

Ultra-relativistic particles have γ significantly larger than 1. Hence:

$$\tanh y \approx \left(1 - \frac{1}{2\gamma^2}\right)\tanh\eta. \tag{2.13}$$

Unfortunately the tanh function is asymptotic and this does not translate to a direct relationship, or even a bound in the relative error $\frac{|y-\eta|}{y}$. Indeed, the longitudinal rapidity is bounded

$$y = \frac{1}{2}\ln\left(\frac{E+p_z}{E-p_z}\right) = \ln\left(\frac{E+p_z}{m_T}\right) < \ln\left(\frac{2E}{m_T}\right) < \ln\left(\frac{2E}{m_0}\right) = \ln(2\gamma), \tag{2.14}$$

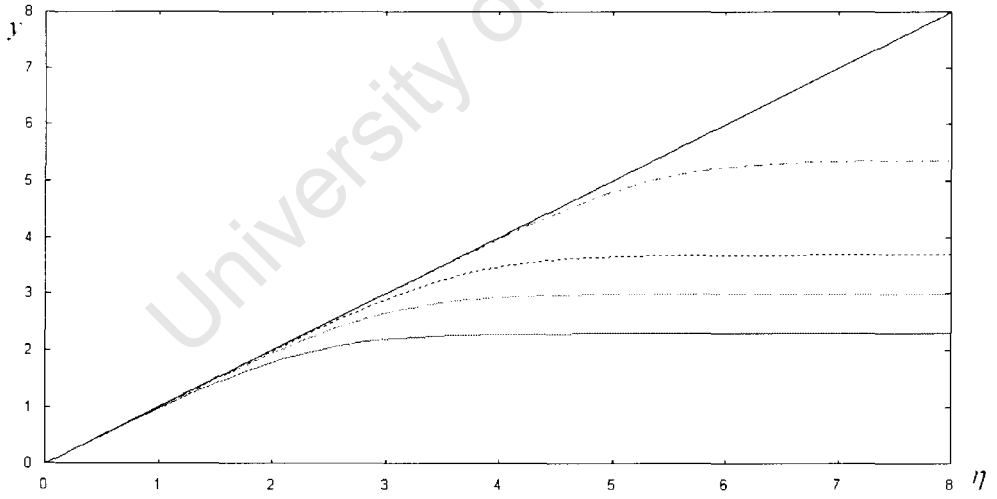

Figure 2.2: Plot showing the rapidity of a particle based on its observed pseudorapidity. The curves correspond to $\gamma = 5$ (solid), $\gamma = 10$ (dashed), $\gamma = 20$ (dots) and $\gamma = 106.5$ (chained - maximum RHIC beam energy), while the straight line is $y = \eta$ for massless particles i.e. $\gamma = \infty$.

whereas η can tend to infinity (for particles where $p = p_z$). However, if the particles are relativistic, it can be seen from figure 2.2 that $y \approx \eta$ in the central rapidity region (up to $\eta \sim 2$)[4].

A different approach to finding the y-η correspondence comes from an observation that the energy of a particle is not independent of the direction in which it is emitted. Since the beam energy is being very high, only the transverse component of the momentum is produced in the collision - the longitudinal component is (to a good approximation) inherited completely from the original colliding nuclei. This 'transverse' approach is assisted by the relation:

$$p_z = m_T \sinh y = p_T \sinh \eta. \tag{2.15}$$

Thus armed, it is possible [4] to find an expression for the 'shift':

$$\delta\eta \equiv \eta - y = \frac{1}{2} \ln \left(1 + \frac{m^2}{p_T^2} \right) + \ln \left(\frac{1 - e^{-2y}}{1 - e^{-2\eta}} \right). \tag{2.16}$$

As $y < \eta$, it follows that $1 - e^{-2y} < 1 - e^{-2\eta}$ and the second term in the expression is negative. At central rapidity it has been shown (via figure 2.2) that $y \approx \eta$, and hence the focus will now turn to forward rapidity. Consider the region where $\eta > 2$, and thus $e^{-2\eta} < 0.02$. The second term becomes

$$\ln \left(\frac{1 - e^{-2y}}{1 - e^{-2\eta}} \right) = \ln \left(1 + \frac{e^{-2\eta} - e^{-2y}}{1 - e^{-2\eta}} \right) \approx \frac{e^{-2\eta} - e^{-2y}}{1 - e^{-2\eta}} \approx e^{-2\eta} - e^{-2y}. \tag{2.17}$$

which can be rewritten in terms of $\delta\eta$ as

$$\ln \left(\frac{1 - e^{-2y}}{1 - e^{-2\eta}} \right) \approx e^{-2\eta} \left(1 - e^{2\delta\eta} \right), \tag{2.18}$$

and therefore, for forward rapidities, the 'shift' satisfies the relation

$$\delta\eta = \frac{1}{2} \ln \left(1 + \frac{m^2}{p_T^2} \right) - e^{-2\eta}(e^{2\delta\eta} - 1). \tag{2.19}$$

The upper bound for $\delta\eta$ is the first term of the expression (2.16). This depends on the quantity of transverse momentum which the particle picked up during the collision, and clearly, if $p_T = 0$, then $\delta\eta \to \infty$. If one imposes a minimum cut on the transverse momentum (which the experiments usually do), one eliminates the infinity problem. The maximum allowable values of the 'shift' can be calculated for each particle species as a function of p_T.

[4]I'd like to thank Dr. Tadeusz Stankiewicz for his assistance with this analysis.

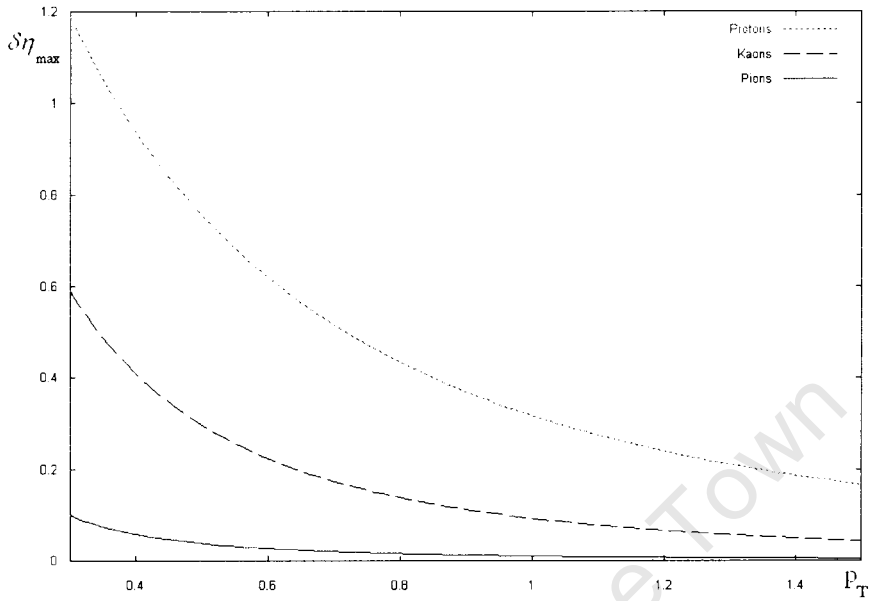

Figure 2.3: The leading term of the $y - \eta$ shift (2.16) for pions, kaons and protons for a range of transverse momenta $0.3 < p_T < 1.5$ GeV.

It is also worth noting that this is the first term of the shift, and that there is also a negative term which scales as $e^{2\delta\eta}$. As such, the shift is smaller than plotted in figure 2.3. It is clear that for pions, which contribute about 80% of the particle spectrum, given a minimum cut of $p_T = 0.3$ GeV, the distribution in y and η will be virtually identical. The main change comes from the heaviest particles (which are mainly nucleons). In this case, a definite correction needs to be made. Although to get the correct rapidity spectrum from a collision one has to measure the transverse momenta of all particles, the baryonic spectrum will be significantly narrower in rapidity than in pseudorapidity by between 0.5 and 1 units, depending on the cuts.

The total hadronic spectrum $\frac{dh}{dy}$ will have only a small change from the spectrum in pseudorapidity due to the domination of the light particles. The plateau observed for a wide central η region [24, 25] will translate into a similar plateau in rapidity, giving weight to the Bjørken scenario (discussed later).

2.3 Ideal hadron gas model

A heavy ion collision, after a certain initial evolution, can be well represented as a gas of hadrons and heavy resonances. It is the properties of this 'gaseous' state which determine which hadrons are emitted from the system[5] and finally detected.

2.3.1 Theoretical formulation

The "ideal hadron gas model" treats the fireball as a strongly interacting ideal gas. At freeze-out the hadrons are expected to be in local thermal equilibrium. As the system is strongly interacting and the particle interactions are allowed to be inelastic (so particles can be produced and annihilated), for a statistical treatment one has to use the Grand Canonical approach. First consider a *static* system (constant volume) in thermal and chemical equilibrium.

For every type of particle in the system (hadron of species i), the mass m_i and spin degeneracy $g_i = 2J_i + 1$ are known. The momentum distribution function is

$$f_i(p) = \frac{1}{(2\pi)^3} \frac{1}{e^{\frac{E_i(p)-\mu_i}{T}} \pm 1} = \frac{1}{(2\pi)^3} \left[e^{\frac{\sqrt{p^2+m_i^2}-\mu_i}{T}} \pm 1 \right]^{-1}, \tag{2.20}$$

where μ_i is the chemical potential, and the loose term is $+1$ is for fermions and -1 for bosons. Hence the number of hadrons of type i in the system is given by

$$N_i = V n_i = V g_i \int f_i(p) d^3p, \tag{2.21}$$

where n_i is defined as the 'number density'.

The quantity which is, as yet, not well defined is the chemical potential μ_i. The system is strongly interacting, which implies that the conserved quantities are those that are not violated in any strong interaction. These include baryon number, charge, strangeness and charm. By assigning a local chemical potential to each of these quantities, we can define the total chemical potential μ_i of a particle as a linear combination of the individual potentials:

$$\mu_i = \mu_B \mathtt{B}_i + \mu_Q \mathtt{Q}_i + \mu_S \mathtt{S}_i + \mu_C \mathtt{C}_i, \tag{2.22}$$

[5] In this model, it is customary to use the term 'fireball' for the system.

where $\mathtt{B}_i$, $\mathtt{Q}_i$, $\mathtt{S}_i$ and $\mathtt{C}_i$ are the baryon number, charge, strangeness and charm of the i^{th} hadron respectively. Sometimes charm is omitted from this formalism as it only effects the very heavy hadrons. This linear relationship follows from the requirement that the chemical potential of any closed system must remain constant under the allowed reactions. It is trivial to show that this is satisfied here.

It is important to note that the quantities μ_B, μ_Q, μ_S and μ_C are not independent. As the system is strongly interacting and there was no strangeness or charm in the initial state, there can be no net strangeness or charm in the final state of the hadron gas. To satisfy this, one must have:

$$\sum_i \mathtt{S}_i N_i = V \sum_i \mathtt{S}_i n_i = 0 = V \sum_i \mathtt{C}_i n_i. \tag{2.23}$$

Furthermore, the system must have the same net charge and baryon number as the initial state (the colliding nuclei). Individually these conditions are tricky to work with as they both depend on the volume of the system, but their ratio does not:

$$\left(\frac{\sum_i \mathtt{Q}_i N_i}{\sum_i \mathtt{B}_i N_i}\right)_{HG} = \left(\frac{\sum_i \mathtt{Q}_i N_i}{\sum_i \mathtt{B}_i N_i}\right)_{nuclei} = \frac{Z}{A}. \tag{2.24}$$

Thus, if one picks a certain μ_B and T, all the other quantities become well defined (although they have to be worked out using a numerical scheme as the conditions turn out to be rather tedious, being summations over all considered hadrons). The evolution of a system is often shown on a T-μ_B phase diagram. This, however, requires the system to be in equilibrium.

This assumption of general equilibrium is rather unfeasible in a system such as the one created in heavy ion collisions, but has proved exceedingly useful in the fitting of observed particle ratios. For two particle species i and j:

$$\frac{N_i}{N_j} = \frac{g_i \int f_i(p) d^3p}{g_j \int f_j(p) d^3p}. \tag{2.25}$$

The particle numbers are clearly invariant under all transformations and it turns out that this formalism gives the same results for non-static systems as well, provided that the temperature at freeze-out is the same (which is usually true by default) and the baryonic chemical potential remains the same throughout the system. However, μ_B is not constant in various regions of the system if it exhibits (for instance) Bjørken-like behaviour. The model does continue to provide good results in regions that are sufficiently homogenous, so the mid-rapidity region in the Bjørken scenario can still be well described, but cannot be successfully applied to the fragmentation regions.

2.3.2 Particle spectra

In an evolving fireball, there exists a certain critical condition (usually characterized by the temperature) at which the particles are emitted. We assume that at freeze-out the hadrons are in local thermal equilibrium and also that the freeze-out condition is common to all particles.

For a particle of species i emitted at freeze-out, the invariant momentum spectrum is given by the Cooper-Frye formula (see equation (B.6)) [10]:

$$E\frac{d^3N_i}{d^3p} = g_i \int_\sigma f_i(x,p)p^\mu d\sigma_\mu, \tag{2.26}$$

where g_i is the spin degeneracy factor and $f(x,p)$ is the locally defined momentum distribution function:

$$f_i(x,p) = \frac{1}{(2\pi)^3}\frac{1}{e^{(p^\mu u_\mu(x)-\mu_i(x))/T(x)}\pm 1}. \tag{2.27}$$

Here p^μ and $u_\mu(x)$ are the particle's 4-momentum and local 4-velocity of the fireball, $\mu_i(x)$ is the local chemical potential and the loose term is $+1$ for fermions and -1 for bosons.

The freeze-out surface σ is a 2+1 dimensional hypersurface in space-time defined by some freeze-out criterion (discussed in more detail in chapter 3). If the surface is parameterized by three orthogonal co-ordinates, the surface normal is given by [13]:

$$d\sigma_\mu \equiv -\epsilon_{\mu\nu\lambda\rho}\frac{\partial\sigma^\nu}{\partial u}\frac{\partial\sigma^\lambda}{\partial v}\frac{\partial\sigma^\rho}{\partial w}du\,dv\,dw, \tag{2.28}$$

where $\epsilon^{\mu\nu\lambda\rho} = -\epsilon_{\mu\nu\lambda\rho}$ is the totally antisymmetric Levi-Civita symbol.

This form is, however, very unfriendly to work with and it will generally be avoided. If one is working with a specific case in which the freeze-out surface σ has a simple geometrical interpretation (and by extension the whole system must evolve in a simple geometrical manner), the Cooper-Frye integral can often be evaluated analytically. The simplest case is that of a stationary, thermalized system.

2.3.3 Spectra from a stationary source

For a stationary source at a certain fixed temperature T, the freeze-out surface element will be simply

$$d\sigma_\mu = \left(d^3x, \vec{0}\right). \tag{2.29}$$

Starting off with the Cooper-Frye formula (B.6) for a particle species i, and using the differential element equation (A.27), we obtain

$$\frac{d^3N_i}{dy\ p_T\ dp_T\ d\varphi} = g_i \int_\sigma f_i(x,p) p^\mu d\sigma_\mu. \tag{2.30}$$

The system will undergo an *instantaneous* freeze-out. As every volume element of the fireball undergoes a causally independent freeze-out from every other, the particle emissions will be completely isotropic and we can integrate out the φ component. If we assume that the chemical potential $\mu_i(x) = \mu_i$ is position independent, then the integrand becomes

$$f_i(x,p)p^\mu d\sigma_\mu = \frac{1}{(2\pi)^3}\frac{1}{e^{(p^0-\mu_i)/T}\pm 1}p^0 d^3x = \frac{1}{(2\pi)^3}\frac{E}{e^{(E-\mu_i)/T}\pm 1}d^3x, \tag{2.31}$$

and the integration becomes trivial. If the fireball has volume V:

$$\frac{d^2N_i}{dy\ p_T\ dp_T} = 2\pi g_i \int_V \frac{1}{(2\pi)^3}\frac{E}{e^{(E-\mu_i)/T}\pm 1}d^3x = \frac{g_i V}{(2\pi)^2}\frac{E}{e^{(E-\mu_i)/T}\pm 1}. \tag{2.32}$$

From here, one can (theoretically) calculate the rapidity and transverse momentum spectra individually. These require the integration of the right hand side, which does not have solutions in terms of finite quadratures. Progress can, however, be made if one assumes Boltzmann statistics (ignores the ± 1 term). With this approximation:

$$\frac{d^2N_i}{dy\ p_T\ dp_T} = \frac{g_i V}{(2\pi)^2}\ E\ e^{-(E-\mu_i)/T}. \tag{2.33}$$

Rapidity spectra

To obtain the rapidity spectrum $\frac{dN_i}{dy}$ for particle species i, we need to integrate over the transverse factor. It turns out to be significantly easier to integrate through m_T rather than p_T. Specifically:

$$\frac{dN_i}{dy} = \int \frac{d^2N_i}{dy\ m_T\ dm_T} m_T\ dm_T = \frac{g_i V}{(2\pi)^2}\ \int_{m_i}^\infty m_T E e^{-E/T} e^{\mu_i/T}\ dm_T, \tag{2.34}$$

where the integrand runs over the possible values of the transverse mass and, since $m_T \geq m_i$ by definition, the limits of integration are m_i and infinity. Now changing variables, substitute $E = m_T \cosh y$:

$$\begin{aligned}\frac{dN_i}{dy} &= \frac{g_i V}{(2\pi)^2} e^{\frac{\mu_i}{T}} \int_{m_i}^\infty m_T^2 \cosh y\, e^{-\frac{m_T \cosh y}{T}} dm_T \\ &= \frac{g_i V}{(2\pi)^2} e^{\frac{\mu_i}{T}} \cosh y \int_{m_i}^\infty m_T^2 e^{-\frac{m_T \cosh y}{T}} dm_T.\end{aligned} \tag{2.35}$$

The $\cosh y$ term is independent of m_T and hence a constant of the integral. Observe that, by introducing a new variable $\zeta \equiv \frac{m_T}{T}$, so that $dm_T = Td\zeta$, the integral becomes:

$$\frac{dN_i}{dy} = \frac{g_i V T^3}{(2\pi)^2} e^{\frac{\mu_i}{T}} \cosh y \int_{m_i/T}^{\infty} \zeta^2 e^{-\zeta \cosh y} d\zeta. \tag{2.36}$$

The integral can now be handled through a double integration by parts:

$$\int \zeta^2 e^{-\zeta \cosh y} d\zeta = e^{-\zeta \cosh y} \left(\frac{\zeta^2}{-\cosh y} - \frac{2\zeta}{+\cosh^2 y} + \frac{2}{-\cosh^3 y} \right), \tag{2.37}$$

and, as the integrand goes to zero at infinity, one finds that

$$\frac{dN_i}{dy} = \frac{g_i V T^3}{(2\pi)^2} e^{\frac{\mu_i}{T}} e^{-\zeta \cosh y} \left[0 - \left(-\frac{\zeta^2}{\cosh y} - \frac{2\zeta}{\cosh^2 y} - \frac{2}{\cosh^3 y} \right) \Bigg|_{\zeta = \frac{m_i}{T}} \right], \tag{2.38}$$

which, after resubstitution, gives the rapidity spectrum:

$$\frac{dN_i}{dy} = \frac{g_i V}{(2\pi)^2} T^3 e^{-m_i \cosh y/T} \left[\frac{m_i^2}{T^2} + \frac{m_i}{T} \frac{2}{\cosh y} + \frac{2}{\cosh^2 y} \right] e^{\mu_i/T}. \tag{2.39}$$

For massless particles, this reduces to

$$\frac{dN_i}{dy} = \frac{2 g_i V}{(2\pi)^2} T^3 e^{\frac{\mu_i}{T}} \times \operatorname{sech}^2 y. \tag{2.40}$$

We see that the temperature T and chemical potential μ_i are simply scaling factors and do not effect the width of the distribution: $\Gamma_{FWHM} \approx 1.76$ units – the width of the sech^2 function. For heavier particles (especially for baryons) the distribution is even narrower. This is, however, in complete disagreement with the observed widths at the AGS and SPS experiments, where the rapidity widths for the baryons are more than twice the widths predicted for a stationary thermal source. Attempts can be made to fix this anomaly by including not only the spectra from particle species i, but also the feed-down from higher resonances that decay into that species. A full description is provided in [2].

Another interesting aspect, which becomes apparent on closer inspection, is that the rapidity distributions have different ranges depending on the masses of the particles in question. When one plots out the rapidity distributions for pions, kaons and protons (see figure 2.4), it becomes apparent that the particle multiplicities are no longer in proportion. The particle abundances, which come from the static fireball and have been used to fit the global particle ratios [3], cannot be used locally.

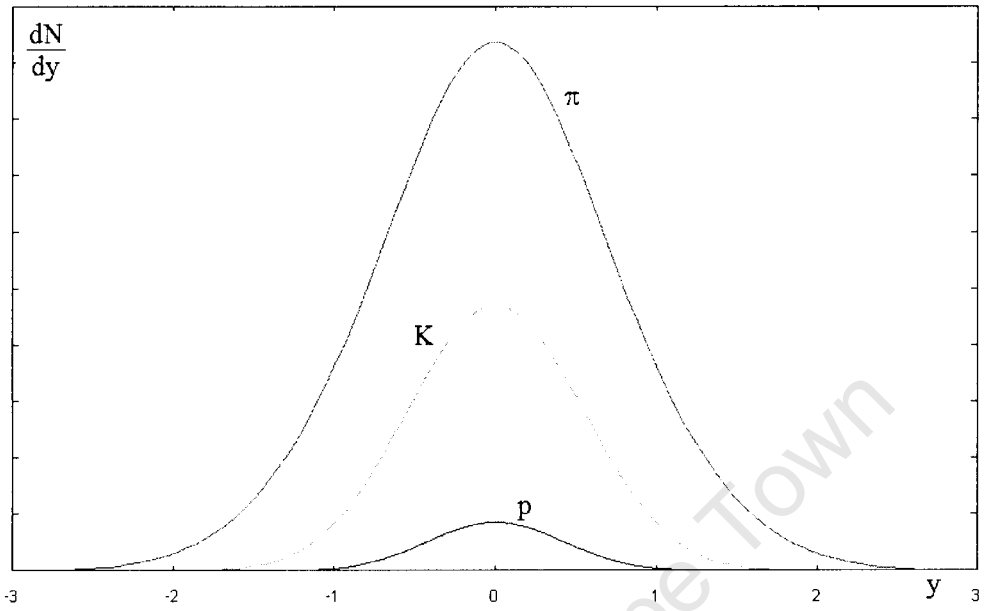

Figure 2.4: Plot showing the rapidity spectra of pions, kaons and protons, as predicted for a stationary thermalized source (2.39). Although the vertical scale is arbitrary (this plot was generated with $T = 170$ MeV and $\mu_i = 0$), observe the discrepancy in particle-ratios at various rapidity regions.

The effect of resonance decays serves to broaden the spectra of the particles [2], but, as the lightest particles (pions) have the most feed-down, this will cause an even greater discrepancy between the particle ratios at mid and forward rapidities. This implies that data taken at mid-rapidity is not representative of the data over the full rapidity range.

One can consider the case of a superposition of individually 'stationary' fireballs, moving at a speed relative to each other, and this can be used to recreate the observed experimental spectra. A flat distribution of fireballs, each with the same local temperature, will ensure that particle ratios remain constant in the central rapidity region (not just at mid-rapidity), but as one moves to the extrema of the system, the particle ratios will again not correspond to those at mid-rapidity. Although this model can be built upon in order to reproduce the experimental observables, it is a better idea to start considering a more realistic system and start examining collective flow – this approach was originally suggested by Landau [7].

2.4 Hydrodynamics

The original analysis of heavy ion collisions at the AGS dealt simply with the measurements of particle multiplicities and distributions. As the energies of the accelerators increased and the energy granted to the system became large enough for the system to sustain itself for a time before hadronization, the idea of collective flow was born. This has since triggered interest in describing the reaction region using a hydrodynamical framework.

A requirement for the application of hydrodynamics is local equilibrium. In (ultra)relativistic heavy ion collisions, this is only reached after some finite thermalization time, and it relies on sufficient rescattering occurring amongst the initial collision products.

If the equation of state (EoS) of the matter under investigation is known, the hydrodynamic equations completely describe the evolution of the system, right up to the point of thermal freeze-out (when the conditions of local thermal equilibrium break down). If the system is not completely well behaved (is chemically unequilibrated or is in a mixed phase), an additional constraint is required to determine the evolution. The most common choice at this point is to impose entropy conservation (as in ideal hydrodynamics).

2.4.1 Relativistic hydrodynamics

In a hydrodynamical description of the evolution of matter, instead of dealing with the individual particles involved (as the transport models do), one considers the flow of particles in a volume element. The equations that govern the evolution are those of conservation of energy-momentum. In addition, the conservation of baryon number is usually implemented. Other conserved currents such as charge, strangeness etc. can also be added.

Starting off with an ideal fluid (no dissipative effects and no viscosity), consider a fluid element from the point of view of a co-moving observer (which is equivalent to observing the fluid element at rest). The energy-momentum tensor is simply

$$T^{\mu_0\nu_0} = \mathrm{diag}(\epsilon, P, P, P). \tag{2.41}$$

To obtain the general form, we need to boost from the co-moving frame to the 'stationary' frame. Hence we Lorentz boost back by the co-mover's relative velocity $u_\mu(x)$, and obtain

$$T^{\mu\nu}(x) = \Lambda^{\mu}_{\mu_0}\Lambda^{\nu}_{\nu_0}T^{\mu_0\nu_0} = (\epsilon + P)u^{\mu}(x)u^{\nu}(x) - Pg^{\mu\nu}. \tag{2.42}$$

Here $\epsilon = \epsilon(x)$ and $P = P(x)$ are the energy density and pressure at space time point $x = x^\mu = (t, x, y, z)$ respectively. Also

$$\frac{dx^\mu}{d\tau} \equiv u^\mu(x) = \gamma(1, \vec{v}), \qquad \text{and} \qquad \frac{dt}{d\tau} \equiv \gamma = \frac{1}{\sqrt{1 - v^2}}. \tag{2.43}$$

The space-time tensor $g^{\mu\nu}$ is the flat Minkowski metric, or explicitly:

$$g_{\mu\nu} = g^{\mu\nu} = \eta^{\mu\nu} = \text{diag}(1, -1, -1, -1). \tag{2.44}$$

To move away from the ideal fluid case to a more accurate representation, one can add features such as bulk and shear viscosity or heat conductivity – these can all be introduced by including their corresponding terms in the energy-momentum tensor (2.42). These terms, however, lead to a large number of additional parameters in the equations of motion and obscure the simplicity of hydrodynamics. As these additional effects are expected to be very small indeed [28] and may well fall within the realm of minor corrections, they will not be discussed here. Fuller analysis are available in [5] and [12].

For energy and momentum to be conserved, a necessary and sufficient condition is that the 4-divergence of the energy-momentum tensor must be identically zero. This gives four coupled differential equations:

$$\partial_\mu T^{\mu\nu} = 0. \tag{2.45}$$

One of these equations is of particular interest here. Substituting for the known form of the energy-momentum tensor (2.42) and contracting with u_ν, the equation becomes

$$u^\mu \partial_\mu \epsilon + (\epsilon + P)\partial_\mu u^\mu = 0. \tag{2.46}$$

Recalling the Euler relation between the 4-divergence ∂_μ and proper time τ of a local fluid element

$$\frac{d}{d\tau} = u^\mu \partial_\mu, \tag{2.47}$$

we find that (2.46) can be rewritten as

$$\frac{\epsilon + P}{\epsilon} \partial_\mu u^\mu = -\frac{1}{\epsilon}\frac{d\epsilon}{d\tau}. \tag{2.48}$$

For $P = 0$, this becomes the continuity equation for the energy density ϵ. The introduction of a non-zero pressure P implies that the energy flow ϵu^μ is no longer conserved. For $P > 0$, there is a transfer of energy into a kinetic form - the matter expands and cools. If $P < 0$ (which might occur during

a mixed phase), the energy is transferred back from kinetic (flow) to the intrinsic energy density ϵ.

There are also additional conserved quantities to consider. In the case of an ideal fluid, these conservation laws can be expressed as the local conservation of the corresponding 4-currents:

$$\partial_\mu(\rho u^\mu) = 0, \tag{2.49}$$

where ρ is the scalar density of the conserved quantity. Expanding this equation gives

$$\partial_\mu(\rho u^\mu) = \rho\partial_\mu u^\mu + u^\mu\partial_\mu\rho = 0, \tag{2.50}$$

which, using relation (2.47), can be transformed into the continuity equation

$$\partial_\mu u^\mu = -\frac{1}{\rho}\frac{d\rho}{d\tau}. \tag{2.51}$$

When dealing with relativistic heavy ion collisions and strongly interacting matter, the net baryon number, charge and strangeness (with densities ρ_b, ρ_q and ρ_s) are conserved, and this gives three specific conservation equations:

$$\partial_\mu(\rho_b u^\mu) = 0, \qquad \partial_\mu(\rho_q u^\mu) = 0, \qquad \partial_\mu(\rho_s u^\mu) = 0. \tag{2.52}$$

As there is originally no net strangeness in the system, there are no strangeness currents to consider, and with $\rho_s = 0$ satisfying both the initial condition and the evolution equation, it can be safely neglected.

Although there is a net (positive) charge in the system initially, the charge current is usually neglected in the treatments as there exists a 'trivial' relation

$$\rho_q = \frac{Z}{A}\rho_b \tag{2.53}$$

to baryon conservation. If the baryon current gets conserved, a trivial solution for ρ_q exists. As such, only baryon conservation needs to be explicitly implemented.

Altogether there are five equations of motion which govern the time evolution of the fluid: four equations in (2.42) and $\partial_\mu(\rho_b u^\mu) = 0$. There are also six fields: energy density, baryon density, pressure and the three independent components of the velocity field. Once a thermodynamic potential is established for the system, the energy density, pressure and baryon density become related by the equation of state. It is from this EoS that macroscopic parameters such as entropy and temperature are derived.

2.4.2 Special cases - dimensional reduction

Any calculation in a full 3+1 dimensional hydrodynamical evolution is a complex process, but for a system such as heavy ion collisions in which the initial conditions are themselves complex, one is forced to resort to a numerical implementation. This requires a rather extensive computer program and lots of computing time, and thus has only become a feasible option in the last decade or so. Before that, the solutions were completely analytical and, as such, only existed for very special initial conditions (such as spherical expansion) which reduced themselves to lesser dimensions through symmetries or the decoupling of longitudinal and transverse flow.

The theorists have been working with reduced dimensions for a significant time, starting with Landau's pioneering work 50 years ago [8]. His work involved a case of 1+1 dimensions, which can be related to a (t, z) system in heavy ion collisions. An analytic solution has been derived for a system which has as initial conditions a stationary square distribution:

$$\rho(0, z) = \begin{cases} \rho_0 & \text{if } |z| \leq a \\ 0 & \text{if } |z| > a \end{cases}, \tag{2.54}$$

for some constants a and ρ_0.

Another approach was started 25 years ago by Bjørken [9]. In this case, the system is modelled as having a very strong flow of matter along the collision axis. It assumes that the colliding particles have so much energy that the flow of energy remains along the original collision axis, and that the system is sufficiently large in the transverse plane to make the edge of the system in the transverse direction of little relevance. It can be thought, qualitatively, that the original nuclei pass through each other and create a cylinder of expanding matter. This approach also reduces the dimensions involved.

Nuclear transparency

The Landau and Bjørken models can both be used to describe the longitudinal behaviour of the fluid in question. They represent, however, very extreme viewpoints of the collision: Landau's work assumes a complete stopping of the nuclei, which deposit all their energy in a small space. Hydrodynamical evolution follows from there. The observed particles will then be centered around mid-rapidity in a distribution as derived in (2.39).

The Bjørken scenario assumes that the original nuclei pass straight through each other with very little loss of speed and deposit energy in the interaction

region, which will expand in a longitudinal fashion. The ideal Bjørken fluid will demonstrate a *boost invariance*, so the rapidity distribution will be flat. The net baryon distribution in this case will be zero at mid-rapidity, and will have a small spread around the original rapidities of the nuclei due to fragmentation.

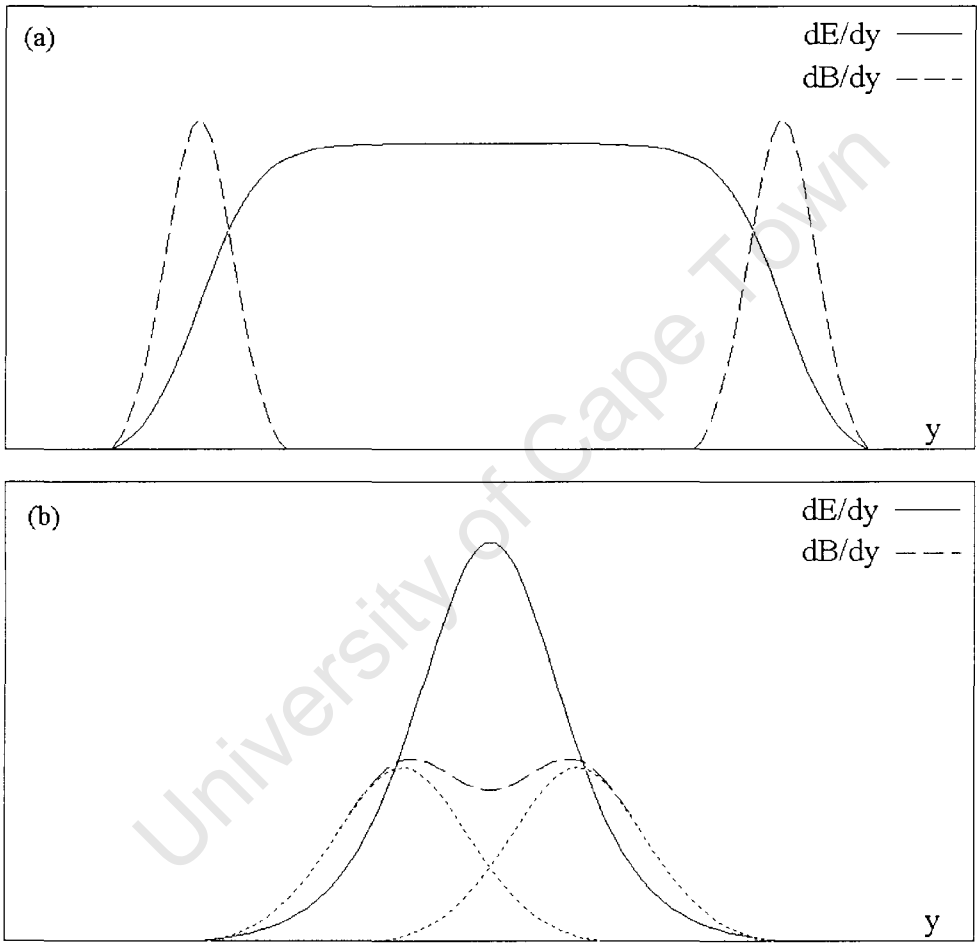

Figure 2.5: Schematic description of the rapidity distribution of energy and baryon number for (a) a fully 'transparent' reaction (Bjørken scenario), and (b) complete stopping (Landau) [4].

The experimental data collected at RHIC in the past years (and to a lesser extent at the SPS) has found that the particle distributions in *pseudorapidity* show a plateau in the central region, which drops off rather steeply at

far forward rapidity [24, 25]. Naïvely one could assume that this plateau would translate to a similar plateau in the rapidity distribution. However, the data obtained at BRAHMS [26] shows that at the highest RHIC energies the individual particle distributions for pions and kaons are very well described by wide Gaussian distributions ($\sigma \approx 2.2$), which gives weight to the Landau scenario. When one considers baryons, the observed rapidity distributions behave very differently and show very definite evidence of nuclear transparency. There is a small net baryon surplus at mid-rapidity, but most of the net baryons are distributed at forward rapidity. The BRAHMS data shows that the average rapidity loss per participant nucleon is 2.0 ± 0.2 units of rapidity [26], and this remains consistent over a broad range of beam energies (from SPS to RHIC energies).

It becomes clear that the complete evolution of the system deviates significantly from either of the idealized models of Bjørken and Landau. However, a fair amount of work has been put into developing a $2+1$ dimensional hydrodynamical evolution for transverse spectra, and, in order to reproduce full spectra, it is not uncommon to attach a longitudinal Bjørken-like behaviour onto this [1]. The Bjørken formalism is assumed to work exactly for the $\sqrt{s_{NN}} \to \infty$ case, and is generally considered to be elegant.

2.4.3 Bjørken hydrodynamics

The results from heavy ion collisions at RHIC [24, 25] (and also to a slightly lesser extent at CERN-SPS experiments) reveal a 'plateau' in the observed particle spectra in pseudorapidity, which translates to a similar plateau in rapidity y. Bjørken [9] suggested a model which replicates this invariance of the central rapidity region under Lorentz transformations. This *boost invariance* reflects a symmetry of the system (assumed to begin at the time of thermalization), which is preserved in the subsequent hydrodynamic evolution of the system. The boost invariance assumption implies that all the fluid elements behave identically in their rest frames; hence all the thermodynamic variables (and thus all observables) can only depend on the proper time τ of each fluid element. Particularly

$$\epsilon = \epsilon(\tau), \qquad P = P(\tau), \qquad T = T(\tau). \tag{2.55}$$

A particle moving at a velocity v_z for a proper time τ has observed time and longitudinal displacement

$$t = \frac{\tau}{\sqrt{1 - v_z^2}}, \qquad z = v_z \frac{\tau}{\sqrt{1 - v_z^2}}, \tag{2.56}$$

which can be combined to give the proper time relation

$$\tau = \sqrt{t^2 - z^2}, \tag{2.57}$$

and the 4-velocity

$$u^\mu = \frac{1}{\tau}(t, 0, 0, z) = (\cosh y, 0, 0, \sinh y). \tag{2.58}$$

Now for this 'special' case:

$$\partial_\mu u^\mu = \frac{\partial u^0}{\partial t} + \frac{\partial u^3}{\partial z} = \frac{\partial}{\partial t}\frac{t}{\tau} + \frac{\partial}{\partial z}\frac{z}{\tau} = \frac{1}{\tau}. \tag{2.59}$$

The equation (2.48) simplifies to the form

$$\frac{\epsilon + P}{\tau} + \frac{d\epsilon}{d\tau} = 0, \tag{2.60}$$

and this determines the evolution of the system, subject to an initial condition

$$\epsilon(\tau_0) \equiv \epsilon_0, \tag{2.61}$$

with τ_0 being the time of thermalization, generally assumed to be reached at a proper time $\tau_0 \equiv 1$ fm/c after the collision [2]. A slightly more in-depth analysis of the Bjørken model will also incorporate the speed of sound in the fluid, which is defined as

$$\frac{1}{v_s^2} \equiv \left.\frac{d\epsilon}{dP}\right|_{S=\text{constant}}. \tag{2.62}$$

In this model it can be shown [4] to obey

$$\frac{v_s^2}{\tau} = -\frac{1}{T}\frac{dT}{d\tau}. \tag{2.63}$$

If the speed of sound changes slowly, this can be directly integrated:

$$T = T_0 \left(\frac{\tau_0}{\tau}\right)^{v_s^2}, \tag{2.64}$$

where T_0 is the temperature at the thermalization time τ_0. Now for a relativistic gas $v_s^2 \leq \frac{1}{3}$, and thus the decrease in temperature is very slow. Indeed, for the system to cool by a factor of 2, the time required is $\tau \approx 8\tau_0$. This is in disagreement with experiment - the system cools significantly faster.

2.4.4 Shortcomings of the Bjørken model

The ideal Bjørken model has been well studied and allows for exact solutions. Although nuclear collisions do exhibit a certain degree of nuclear transparency, there are aspects of this model which do not apply to certain regions of the system's evolution.

Transverse flow is not implicit in the model described above. As transverse flow must exist in the hydrodynamical picture, modifications have to be made. It is not uncommon to describe the collisions with 2+1 dimensional hydrodynamics (x and y evolution), and to superimpose a longitudinal Bjørken flow onto it. Such work has been done numerically [1], as well as analytically for central collisions [2]. However, the data taken at RHIC disagrees with these assumptions [20, 26]. Moreover, these models completely ignore the fragmentation regions of the collision, and assume a constant baryonic potential for all rapidities. Finally, the individual particle ratios at mid-rapidity are not representative of the whole collision, and thus, to have a complete hydrodynamical description of a system, one has to work with the full 3+1 dimensions.

It turns out that for a full hydrodynamical evolution the freeze-out hypersurface is not simply defined. The next chapter will deal with the construction of this hypersurface, using a numerical hydrodynamical scheme.

Chapter 3

Numerical Freeze-out

In numerical hydrodynamics, the continuous fluid has to be discretized onto a finite lattice in both spatial and temporal dimensions. The spatial discretization is usually uniform over the three dimensions, and the time intervals are chosen sufficiently small to allow the system to evolve without too much truncation. For heavy ion collisions one expects a very strong flow in the longitudinal direction ($v \simeq (1 - \frac{1}{2\gamma^2})c \approx c$), and a comparatively slow expansion in the transverse directions. As such, it has been suggested to use a non-uniform discretization with the cells longer in the z-direction than in the transverse directions [39]. However, this break of symmetry leads to complications both in the evolution of the system (extra scaling in the derivatives), as well as in freeze-out development (surface normals become complicated to compute). The work here assumes that the discretization is uniform in the spatial dimensions ($\Delta x = \Delta y = \Delta z = 1$), although it allows for more cells in the longitudinal direction than the transverse ones.

In such a hydrodynamical framework, after discretization, the system will be represented at every time step by a 3 dimensional lattice, with every point referring to a small volume element. Every point will have an associated temperature, energy density, flow and baryon density. It is from this data that the freeze-out surface can be numerically constructed, which is necessary for the determination of observables (spectra).

As freeze-out refers to the time at which hydrodynamical evolution stops, if we assume that the system does not at a later time heat up, the sections which have frozen-out will have no effect on the remaining hydrodynamics. Hence the freeze-out program can be run after the evolution has been completed, and, by variation of parameters, various freeze-out criteria can be compared.

3.1 Freeze-out mechanism

Freeze-out (as mentioned in section 2.4), occurs when the system evolves to a point where the temperature and energy density are too low for thermal equilibrium to be maintained, and the hydrodynamic assumptions no longer apply. Some works [1] use only the energy density, while others rely solely on a critical temperature [2, 3]. Nonetheless, it is generally accepted that the freeze-out criterion is a function of these two parameters. There must exist a continuous function which, at the freeze-out criterion, satisfies

$$\mathcal{F}(\epsilon, T)\Big|_{crit} = 1, \tag{3.1}$$

and takes on values greater than one for a hydrodynamical system and less than one for a frozen-out system. As the function $\mathcal{F}$ is user defined, it can be evaluated at every point on the numerical lattice. This is the parameter which will be used as input for the program.

3.1.1 Shape of freeze-out

There has been a fair amount of theorizing on the behaviour of the freeze-out surface in a hydrodynamical system. In the simplest case - a system with azimuthal symmetry (a fully central collision) and local boost invariance - the system exhibits a cylindrical symmetry. The freeze-out will correspond to a direct relation between the time t and radius r.

In the moments just after the collision, at RHIC energies, the system has a temperature significantly higher than the critical value required for freeze-out. It will expand longitudinally at c and in the transverse direction at a different fixed speed. As the system cools, the edges of the fireball will cool first, and the freeze-out surface will get further away from the edge of the matter. In time, the entire system will cool below the critical temperature. There are three possible cases to consider at this point (shown in figure 3.1).

The first case is that of the system expanding 'naturally', with the edges always being cooler than the interior. As the original distribution (from the Glauber approximation) has the highest energy density at the centre of the system, and this falls off slowly towards the edges, this scenario is thought to best represent the evolution of heavy ion collisions.

The second case involves instantaneous freeze-out, as would be found locally (for every z/t value, in the entire transverse plane) in the ideal Bjørken

model. As the original energy distribution is not homogeneous and there is transverse flow, this would only be possible if the freeze-out criterion had an explicit time dependance. This is clearly unphysical and instantaneous freeze-out will no longer be considered.

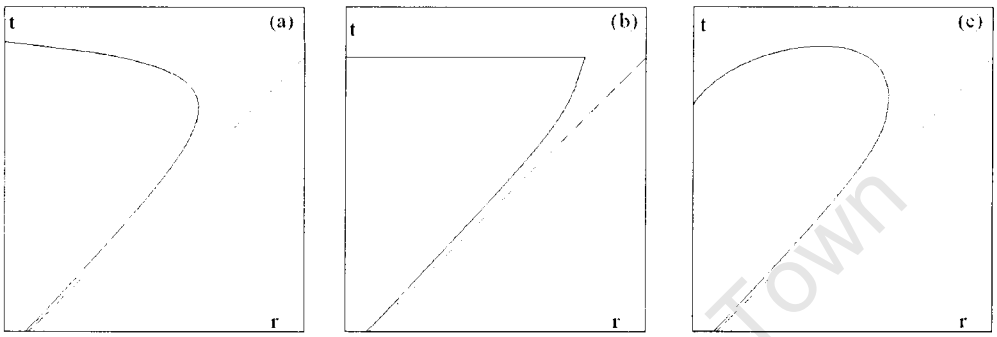

Figure 3.1: The three classical freeze-out scenarios: (a) normal freeze-out from the surface inwards, (b) instantaneous freeze-out and (c) the blast-wave evolutionary model with both inner and outer freeze-out. The dashed line represents the edge of the fluid.

Lastly we consider the case in which the evolution is similar to the 'blast-wave' model. Here the system evolves in such a way as to cause a wave of high energy density to radiate outwards from the centre of the collision, leaving the central region to cool and freeze out before the blast-wave. It remains unclear whether this represents a possible physical scenario. The work done in 2+1 dimensions [1] found no evidence of such behaviour. With this in mind, this model is not directly implemented. Provided is, however, an outline of how such an implementation could be achieved (see section 3.3.1).

3.2 Numerical approach

After the hydrodynamical evolution has been completed, the output required for freeze-out determination is the critical function $\mathcal{F}$ evaluated at every point in the discretized space-time.

To obtain the most accurate hypersurface from the given input, a certain degree of interpolation is required. This can be done either in spatial or temporal dimensions, but not both. The 'simplest' approach follows every

spatial point through the entire evolution, and finds the time at which freeze-out occurs. There are several shortcomings of this method: combining the points of constant freeze-out time is non-trivial and, if one implements the interpolation in time, the freeze-out points will all have slightly different times of freeze-out, resulting in the constructed surface not satisfying (3.1) exactly.

A better approach is to keep the discretized times and interpolate in the three spatial dimensions. In the description of heavy ion collisions, the system will evolve very strongly in the longitudinal direction. The freeze-out surface (transverse contour) will remain similar at neighbouring longitudinal z values, and exploiting this property will result in reduced truncation and a more accurately determined surface.

The approach which I undertook was to look at the hydrodynamical output for constant times t. From there, consider a fixed longitudinal displacement z. This gives a 2 dimensional lattice corresponding to the transverse plane. Contours of $\mathcal{F} = 1$ can be found. By considering the contours for all z values for a constant time, a 2+0 surface can be constructed by triangulation. These surfaces then need to be connected for neighbouring times to form the hypersurface, and only then can the spectra be reconstructed.

The sample code provided has been written in C/C++. Various languages were considered before implementation started and, although a lot of the current theoretical work is still done using Fortran compilers, I have decided not to go that route. People entering the field will (at this point in time) be more familiar with the Java/C/C# style than Fortran, and the aim is to make the code acceptable to the widest possible class of people. The code provided comes with no licence and can be used/canibalised at will.

3.2.1 Contour determination

The starting point in the determination of a hypersurface is to dimensionally reduce it to a case where we're dealing with one dimension. The most logical choice is to consider the transverse plane and find the contour. The input for the program is the critical function $\mathcal{F}$ at every lattice point. It is then converted into a binary function `Crit` as

$$\texttt{Crit[x][y][z]} = \begin{cases} 0 & \text{if } \mathcal{F}(x,y,z) < 1 \\ 1 & \text{if } \mathcal{F}(x,y,z) \geq 1 \end{cases} . \tag{3.2}$$

The edges of the lattice must be frozen-out for the program to work. Otherwise the freeze-out falls outside the bounds of the lattice and determining the complete surface becomes impossible.

By doing a linear search through the `x` and `y` values, one can find the first lattice point above the critical value. As the point directly before it must be below the critical value (a previous point must exist), through linear interpolation one can determine the first contour point:

```
ContX[0]  =  (T_C-F[X0][Y0])/(F[X0+1][Y0]-F[X0][Y0]) + X0;
ContY[0]  =  Y0;                                                    (3.3)
```

where `T_C` represents the critical value - taken to be 1 (this leaves room for modification). After the first contour point is determined, the rest can be found in a relatively simple manner. By storing the co-ordinates of the last 'cold' and 'hot' points (below and above the critical value), as well as their relative direction, the entire contour can be found.

Defining clockwise to be the preferred direction, the latest contour point lies directly between the last 'cold' and 'hot' points (represented as $-$'s and $+$'s when on a lattice). Next we consider the adjoining lattice points:

- (3+'s and 1$-$) If the point anticlockwise of the 'hot' point is 'hot' as well, then the 'cold' point remains constant and the contour takes an anti-clockwise turn.
- (2+'s and 2$-$'s) If the point anticlockwise of the 'hot' point is 'cold', but the diagonally across point is 'hot', both defining points move across by 1 unit. The contour moves 'straight'.
- (1+ and 3$-$'s) If both the points (anticlockwise and diagonally across) are 'cold', then the 'hot' point remains unchanged, and the contour takes a clockwise turn.

At every step the contour element is found through linear interpolation. Every point on the specific contour will have one of `x` and `y` as integers. The clockwise construction will end when the original point is finally reached. The length of this contour will vary, but it will not generally exceed $4 \cdot$ `C_x`, where `C_x` is the number of cells in each transverse direction (that is the maximum number allowed for a convex contour - concave contours can have an almost unbounded length, but the level of concavity is not particularly high here [1]). Leaving room for special cases, it is a good idea to keep the contour array size at $5 \cdot$ `C_x`, unless memory restrictions intercede.

3.2.2 Surface determination

Once the contours are completed for every value of z, the wire-frame schematic of the surface is known. The next step is to complete the surface by converting the contour elements into surface elements. The natural approach is through triangulation, as triangles are the only polygonal figures that are always confined to a plane. The triangles will be constructed between the contours, with all the vertices defined with integer z values.

One of the major obstacles comes in matching up the neighbouring contours so as to describe the surface accurately. The natural naïve approach would be to take every contour segment as a triangular base and find the closest point (minimum Euclidian distance) on the adjoining contour. This does not, in fact, give a good surface. A simple cylindrical surface with its axis not parallel to the z-axis will give elliptical contours in the transverse plane, and a closest-point triangulation will skew the surface significantly. However, if one considers one of the extreme points on each contour, the implicit similarity at neighbouring longitudinal contours will ensure that these points will correlate to each other on the surface.

Another complication that arises is when the contours have a different number of constituent points (this will virtually always be the case, unless the contours happen to be almost identical). At this point I use the fact that the length of each segment of the contour is (on average) equal, and, by similarity, the contour expansion resulting in a greater number of segments will be (to a good approximation) uniform over the entire contour. If the two contours have length L_1 and L_2, consider the sequences

$$\{a_n\} \equiv \left\{\frac{1}{L_1}, \frac{2}{L_1}, \ldots, \frac{L_1}{L_1}\right\} \quad \text{and} \quad \{b_n\} \equiv \left\{\frac{1}{L_2}, \frac{2}{L_2}, \ldots, \frac{L_2}{L_2}\right\}. \tag{3.4}$$

A complete triangulation will consist of L_1 triangles with bases on contour one, and L_2 triangles with bases on the second contour. For every pair of consecutive elements in a sequence, find the closest element in the second sequence to the average of the two elements. It turns out that this gives the one of best triangulations for the given set of contours.

There are two ways of storing the individual surface elements, both with their individual applications. The standard way involves storing the co-ordinates of the vertices of every triangle. A secondary way, which causes a slight loss of information, is to store simply a surface element with a position, area and orientation in space. The centroid of every triangle will represent its position, whereas the cross product of (any) two sides will give the normal to

the surface, which uniquely defines the orientation. The length of the normal is arbitrary, and can be scaled by the area of its corresponding triangle. Hence we need store only the position of the centroid and the scaled normal to represent the surface elements. The area is obtained by use of Heron's formula:

$$A = \sqrt{s(s-a)(s-b)(s-c)} = \frac{1}{4}\sqrt{(a+b+c)(a+b-c)(b+c-a)(c+a-b)}, \tag{3.5}$$

where a, b, c are the sides of a triangle and s is the semiperimeter.

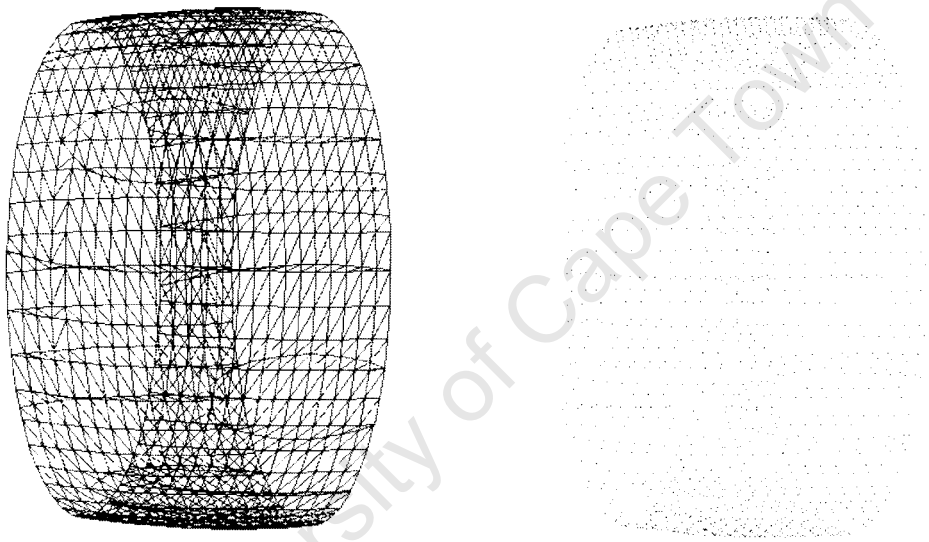

Figure 3.2: Plot showing the two styles of storing individual surfaces: The triangulated approach (left), and using the centroids of the constituent triangles (right), with the scaled normals not shown.

3.2.3 Matching up individual surfaces

The centroid-normal format of storing the surfaces was introduced so as to allow for an efficient way of combining the individual surfaces in time. While it is convenient to store the surfaces without loss of information (triangulated data), there is no convenient way of matching up two such surfaces. As the labelling of the triangles does not preserve surface compactness (neighbouring surface elements cannot have neighbouring labels), it becomes impossible to do a surface matching using a one dimensional variable, as was done in

contour matching. Also, the number of surface elements is expected to vary as the system evolves in time. Hence, if one uses surface elements based on simple interpolation without rescaling, there can be no bijective correlation between the two discrete times.

Consider the centroid-normal representation of a surface at time t. The normal of the surface defines the line:

$$\vec{x}_t + k\,\hat{n}_t \qquad \forall k \in \mathbb{R}. \tag{3.6}$$

At the next time step, each triangular surface element uniquely defines a plane in space. Unless the normal $\vec{n}_t$ is parallel to a plane, there exists a unique intersection point. One can proceed by finding which surface element the line (3.6) passes through, and matching the surfaces in this way. However, finding the numerical formulation of the plane given three points in space, and finding the intersection of a (general) line with such a plane is numerically unfeasible.

A simpler approach is to notice that at the next time interval, every surface element can also be represented in a similar way, with index $t+1$. By finding the centroid at time $t+1$ which is closest to the normal line (3.6), and repeating for all surface elements at time t, the two surfaces can be appropriately matched up. The nearness equation is

$$\vec{D}(k) = (\vec{x}_t - \vec{x}_{t+1}) + k\,\hat{n}_t. \tag{3.7}$$

To obtain a good approximation for the minimum value of $|\vec{D}(k)|$, we first dot the equation with $\hat{n}_t$:

$$\vec{D}(k) \cdot \hat{n}_t = D(k)\cos\theta = (\vec{x}_t - \vec{x}_{t+1}) \cdot \hat{n}_t + k, \tag{3.8}$$

and, as we're looking for a case where the normal passes as close as possible to the centroid of its corresponding triangular surface element, $D \approx 0$, which occurs for

$$k \approx (\vec{x}_{t+1} - \vec{x}_t) \cdot \hat{n}_t. \tag{3.9}$$

Hence we just need to find the minimum value of the expression

$$\vec{D}_{min} = (\vec{x}_t - \vec{x}_{t+1}) + [(\vec{x}_{t+1} - \vec{x}_t) \cdot \hat{n}_t]\,\hat{n}_t. \tag{3.10}$$

By looping through all the surface elements of the next time step, the minimum can be found, and hence we can match up every surface element at t to a surface element at $t+1$. There must also be a veto-condition that the normals must point in roughly the same direction (so as not to match up a

surface element with one at the opposite end of the hypersurface). Thus we require

$$\hat{n}_t \cdot \hat{n}_{t+1} \geq \frac{1}{\sqrt{2}}, \tag{3.11}$$

which ensures that the normals do not form an angle greater than 45°.

3.2.4 Spectra reconstruction and $(p^\mu \sigma_\mu)$

The Cooper-Frye formalism involves integrating the particle flow through the freeze-out surface. There have been proposed modifications to this, most notably the θ-modified Cooper-Frye form, where a theta function is placed around the $p^\mu \sigma_\mu$ term to eliminate the (unphysical) scenario of particle re-absorption when the freeze-out surface expands faster than the fluid flow. This crude approach has various problems, most notably because it violates energy conservation, but has attracted a fair amount of interest. Other formalisms that conserve energy and are positive-definite have since been suggested [11]. The standard Cooper-Frye form is discussed here.

The flow of particles through the surface is described by the Lorentz covariant term

$$p^\mu d\sigma_\mu, \tag{3.12}$$

which is invariant under boosts. The surface element $d\sigma_\mu$ is algebraically defined in (2.28), but that form is completely unusable in a numerical scheme. We can, however, exploit the boost invariance implicit in (3.12). Recall that for a stationary source (section 2.3.3) the surface integration becomes trivial, and that rapidity is additive under Lorentz boosts (A.18).

As the surfaces are matched up using surface element normals, there is a simple mapping of surface elements onto each other through the given time step. By transferring to a moving frame in which the local fluid element under investigation remains stationary, the integration can be simply done. If the local surface fluid element has associated rapidity y_σ, we can boost the fluid flow p^μ into the surface frame (it becomes p^μ_σ), and obtain

$$\begin{aligned} E\frac{d^3N_i}{d^3p} \sim \int_\sigma f_i(x,p) p^\mu d\sigma_\mu &= \sum_{\delta\sigma} \left(\int_{\delta\sigma} f_i(x,p) p^\mu_\sigma d\sigma_{\mu(rest)} \Big|_{y_\sigma} \right) \\ &= \sum_{\delta\sigma} \left(\int_{\delta\sigma} f_i(x,p) p^0_\sigma d^3x \Big|_{y_\sigma} \right) \\ &= \sum_{\delta\sigma} \left(|\delta\sigma|\, f_i(x,p) p^0_\sigma \Big|_{y_\sigma} \right), \end{aligned} \tag{3.13}$$

where $f_i(x,p)$ and the flow are assumed to be constant over every surface element $\delta\sigma$, and $|\delta\sigma|$ refers to the volume encompassed by the surface elements. From this we can find the rapidity spectra by using (A.27) and integrating out the transverse component:

$$\frac{dN_i}{dy} = \iint \left[E\frac{d^3N_i}{d^3p} \right] m_T \, dm_T \, d\varphi \sim \sum_{\delta\sigma} |\delta\sigma| \left[\int f_i(x,p) p^0_\sigma m_T \, dm_T \, \Big|_{y_\sigma} \right]. \tag{3.14}$$

By carefully calculating the rapidity of the surface elements $\delta\sigma$ (which have a strong dependence on the time-intervals), the rapidity spectra from 3+1 dimensional hydrodynamics can be constructed.

3.3 Areas for improvement

The algorithms described above work very well for an evolution in which the surfaces remain convex, and also for surfaces which have a small degree of concavity. The analysis done by Kolb (figure 2.13 in [1]) shows that for very peripheral collisions ($b \geq 7$ fm), in the initial few moments after thermalization, the contours in the transverse plane show a large deviation from convexity. It becomes a problem when there exist two points along a contour whose surface normals become parallel.

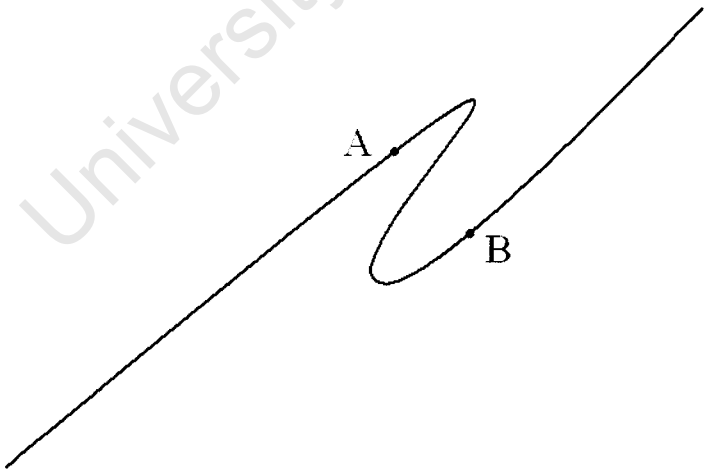

Figure 3.3: The 'catastrophe' which comes about when the freeze-out contour overlaps sufficiently for the normals to become degenerate.

In the reduced case, where one only works in the transverse plane, this is not a problem as the expanding contour can be matched up with the next time step through continuity (a contour has a S topology and a continuous $S \to S$ mapping is trivial). When considering 3+1 dimensions, every individual surface has 2 dimensions and a $S \times \mathbb{R}$ topology. When discretized, this cannot be parameterized by a single discrete variable that preserves surface compactness.

When matching up the individual surfaces, the code uses surface normals to find an adjoining surface element in the next time step. This method works very well, provided that the normal construction results in a single associated surface element. In a case such as is shown in figure 3.3, points A and B will be associated with the same surface element and the constructed hypersurface will fold onto itself, which is highly undesirable as the numerical code will no longer represent the physics in question.

3.3.1 Convex hull completion

To prevent the numerical code from associating the same surface element with distinct, non-adjacent points in the previous time step, several suggestions have been made. One could implement a set 'nearness' flags into the program - by constructing a coarser discretization of the spatial lattice, the surface elements can be bucket sorted by observing which region of space they exist in. By imposing a condition that a surface element can only be matched up to another if they fall into the same or directly adjoining buckets, the 'catastrophe' can be avoided. However, in order to implement this successfully, the bucket sizes need to be very carefully constructed, and probably more than one set of coarse discretizations will be required [37].

Another idea [38] exploits the additive properties of surface integration, and the knowledge that the current algorithms work well for surfaces without major concavities. This idea, termed the convex hull completion algorithm (CHC), involves integrating over a slightly different surface (which is well behaved and does not cause severe complications), and removing the added surface bits later. The term 'convex hull' is a little untrue as one does not have to construct a proper convex hull from the individual contours - slight concavities can be handled normally. Rather it involves closing off only the major problem-causing sections.

When the contours are constructed, one can implement the CHC by checking for major sections of concavity. As the points on a contour are interpolations

between neighbouring lattice points, the length of the contour is linearly proportional (barring some pathological cases) to the number of points in that contour. Once a convex hull has been constructed, every point can be flagged as either forming or being inside the hull. A set of consecutive points which are inside the hull (given some minimum bound) can be excluded from the contour, and used to create a sub-contour (making it go in the anti-clockwise direction). The assumed continuity of the freeze-out in the longitudinal direction will ensure that at neighbouring z values the transverse plane contour will also remain concave and have a similar sub-contour (unless the sub-contour contains too few points to trigger the bound). As such, the constructed surface will not be a single surface, but will be a (mostly) convex surface, with possibly small sub-surfaces. If the sub-contours are constructed anti-clockwise, this will cause the resulting sub-surfaces to be oppositely orientated to the main surface. By running them through the same procedure of surface matching, the opposite orientation will create the relevant spectra as negative, and the results can be simply added to create the final spectrum.

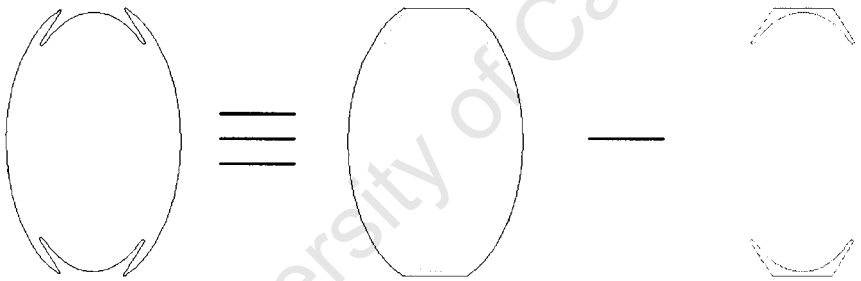

Figure 3.4: Constructing the convex hulls for contours which exhibit the catastrophic properties of figure 3.3. The red lines refer to the new contours of integration, and the grey indicates the sections that are neglected.

It ought to be noted that, for this model to be implemented, the θ-modified Cooper-Frye formulation cannot be used, as it will veto the negative spectra. Also, minor complications can arise if the constructed sub-surfaces show sufficient concavity themselves to cause numerical problems. These are not, however, expected to come about in heavy ion collisions.

Freeze-in surfaces

As was mentioned earlier, there is a discrepancy in the general opinion in the field over whether a hydrodynamical description of heavy ion collisions can

result in the centre of the system cooling to below the freeze-out criterion before the outer edges (figure 3.1 (c)). There is also a method for including this freeze-in surface into the spectra. The method turns out to be very similar to that outlined for the CHC algorithm.

A contour in the transverse plane is found by observing where the critical function $\mathcal{F}$ changes from below to above 1. After the outer contour has been found, one can 'paint' the region outside the contour with a value above the critical. By then modifying the parameter values at every point `Crit[x][y] = 1-Crit[x][y]`, a new set of data can be created. The area outside the freeze-out contour (already found), as well as the previously hot matter will be marked as cold. Indeed, the only region which the code could pick up will be the frozen-out matter inside the hydrodynamical matter. If the freeze-in surface exists, it can be marked with the opposite orientation. To find the complete spectra, one proceeds as above. (Sample code for converting the parameters to obtain the inside contours is given in the functions `Doughnut` and `Paint`.)

3.3.2 Bijective surface mapping

Another step that requires more attention is the surface mapping algorithm. An expanding surface will result in more individual surface elements (as the average size of each surface element will remain constant), and hence the surface matching function cannot result in a surjective mapping. Even injectivity cannot be guaranteed by the 'closest-point' approach. An ideal algorithm will involve a mapping that displays bijectivity (complete functional invertibility). This idea has strong requirements.

For a finite domain, a bijective function will also require a finite, equally-sized co-domain (bijective functions preserve set cardinality). However, a fixed number of surface elements will require their size to scale with the total surface area. The algorithm for finding surface elements (inter-contour triangulation) will have to be strongly modified. As the bijective function would be expected to preserve surface compactness, the surface-mapping approach will have to be completely changed as well.

I have no current knowledge on how such a bijective mapping might be achieved, or even if it is possible.

Chapter 4

Summary and Conclusion

A review of the macroscopic description of heavy ion collisions has been presented, with focus on hydrodynamics, when modelling the evolution of the system, and the ideal hadron gas model for generating spectra. Simplifications in both have been well studied and these ideas presented. They do, however, all show a certain degree of disagreement with experimental observables.

The statistical approach to particle production has been investigated: the idea of static particle-emitting sources, though attractive, was shown to be unfeasible and to disagree with the spectra obtained even for low energy collisions. Superpositions of locally stationary fireballs were also shown to fall short in the single description of local and global particle spectra. We were forced to conclude that hydrodynamical flow is a necessity.

In the framework of hydrodynamics, the dimensionally reduced models have been examined: the Landau model (for nuclear stopping) was shown to be partially representative of the spectra obtained in heavy ion collisions, although the net baryon distributions show complete disagreement. They demonstrate a level of nuclear transparency more applicable to the Bjørken model, which does not, in turn, reproduce the spectra for light particles, and is completely unable to recreate the fragmentation regions. These shortcomings provide strong motivation for studying full 3+1 dimensional hydrodynamical evolutions.

The implementation of a hydrodynamical evolution is beyond the scope of this thesis. However, every hydrodynamical scheme (ideal, viscous, etc.) has a critical condition (freeze-out) at which the fluid assumption breaks down and the particles are emitted from the medium (distributions are usually

prescribed by the Cooper-Frye formula). Freeze-out criteria have been briefly discussed, and a numerical implementation for this freeze-out hypersurface is provided, together with the formalism for generating particle spectra from this surface.

The algorithm provided works well when describing the hydrodynamical evolution of central heavy ion collisions, and remains accurate up to an impact parameter of $b \approx 6$fm, at which point the level of surface concavity becomes too great [1] for the numerical implementation of the surface-normal algorithms to function properly. These effects might disappear once viscosity is implemented [12, 29]. Algorithms for bypassing the concavity problems are also presented. Examples of surface outputs are presented in appendix D.

Appendix A

Kinematic Variables

In this thesis, I have used the standard set of units for the field, most notably:

$$\hbar = c = k_B = 1. \tag{A.1}$$

As ultra-relativistic heavy ion collisions produce particles that have very large longitudinal momenta, provided is a short review of the relevant relativistic mechanics.

Consider a particle with rest mass m_0. In its rest frame, it has 4-momentum

$$p_0^\mu = (E, \vec{p}) = (m_0, 0, 0, 0). \tag{A.2}$$

To find the momentum of the particle in a frame moving with velocity v along the z-axis, one uses the Lorentz transformation:

$$p^\mu = \Lambda^\mu_\nu p_0^\nu, \tag{A.3}$$

where the transformation matrix is given as

$$\Lambda^\mu_\nu = \begin{pmatrix} \gamma & 0 & 0 & \gamma\beta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \gamma\beta & 0 & 0 & \gamma \end{pmatrix} \tag{A.4}$$

with

$$\beta = v/c = v \qquad \text{and} \qquad \gamma = \frac{1}{\sqrt{1-\beta^2}}. \tag{A.5}$$

A.1 Rapidity formulations

The longitudinal rapidity of a particle is defined as

$$y = \frac{1}{2}\ln\left(\frac{E+p_z}{E-p_z}\right) = \frac{1}{2}\ln\left(\frac{(E+p_z)^2}{E^2-p_z^2}\right) = \ln\left(\frac{E+p_z}{m_T}\right). \tag{A.6}$$

However, this form is hardly ever used in theoretical calculations as it turns out that there is a much more convenient form using hyperbolic functions. Observe that

$$\begin{aligned}
\tanh y = \frac{e^y - e^{-y}}{e^y + e^{-y}} &= \frac{\sqrt{\left(\frac{E+p_z}{E-p_z}\right)} - \sqrt{\left(\frac{E-p_z}{E+p_z}\right)}}{\sqrt{\left(\frac{E+p_z}{E-p_z}\right)} + \sqrt{\left(\frac{E-p_z}{E+p_z}\right)}} \\
&= \frac{\left[\sqrt{\left(\frac{E+p_z}{E-p_z}\right)} - \sqrt{\left(\frac{E-p_z}{E+p_z}\right)}\right]^2}{\left(\frac{E+p_z}{E-p_z}\right) - \left(\frac{E-p_z}{E+p_z}\right)} \\
&= \frac{\left(\frac{E+p_z}{E-p_z}\right) - 2 + \left(\frac{E-p_z}{E+p_z}\right)}{\left(\frac{E+p_z}{E-p_z}\right) - \left(\frac{E-p_z}{E+p_z}\right)} \\
&= \frac{(E+p_z)^2 - 2(E-p_z)(E+p_z) + (E-p_z)^2}{(E+p_z)^2 - (E-p_z)^2} \\
&= \frac{4p_z^2}{4Ep_z} \\
&= \frac{p_z}{E}.
\end{aligned} \tag{A.7}$$

Hence we see that longitudinal rapidity can be written in the following forms:

$$y = \frac{1}{2}\ln\left(\frac{E+p_z}{E-p_z}\right) = \tanh^{-1}\left(\frac{p_z}{E}\right). \tag{A.8}$$

At this point it's appropriate to mention the pseudorapidity η. It is defined in a similar way to the longitudinal rapidity y and, using a derivation identical to (A.7), has a more convenient form in hyperbolic functions:

$$\eta = \frac{1}{2}\ln\left(\frac{p+p_z}{p-p_z}\right) = \tanh^{-1}\left(\frac{p_z}{p}\right). \tag{A.9}$$

Here p is the total momentum of the particle and it is clear that η is a purely geometrical quantity related to the angle that the path of the particle makes with the z-axis. The $y-\eta$ correspondence is described in section 2.2.

A.2 Rapidity transformations

Consider again boosting a particle into a frame moving with velocity v in the z-direction. Look at the rapidity y of (a 'stationary' particle in) this frame:

$$y = \tanh^{-1}\left(\frac{p_z}{E}\right) = \tanh^{-1}\left(\frac{\gamma\beta_z m_0}{\gamma m_0}\right) = \tanh^{-1}\beta_z = \tanh^{-1}\beta. \tag{A.10}$$

Hence we find that

$$\beta = \tanh y, \tag{A.11}$$

which when substituted into the definition of γ yields:

$$\gamma = \frac{1}{\sqrt{1-\beta^2}} = \frac{1}{\sqrt{1-\frac{\sinh^2 y}{\cosh^2 y}}} = \cosh y, \tag{A.12}$$

which trivially gives

$$\gamma\beta = \cosh y \cdot \tanh y = \sinh y. \tag{A.13}$$

Combining these relations, one finds that the Lorentz transformation matrix (A.4) can be written in a neat way using the longitudinal rapidity:

$$\Lambda^\mu_\nu = \begin{pmatrix} \cosh y & 0 & 0 & \sinh y \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sinh y & 0 & 0 & \cosh y \end{pmatrix}. \tag{A.14}$$

Now consider a particle of mass m_0 travelling with rapidity y_i. In the current frame, its 4-momentum is

$$p_i^\mu = m_0\gamma_i(1, \vec{v_i}) \qquad \text{and} \qquad v_{zi} = \tanh y_i. \tag{A.15}$$

Boosting this particle into a frame moving with rapidity y purely in the z-direction, one calculates the 4-momentum in the boosted frame to be:

$$p_f^\nu = \Lambda^\nu_\mu p_i^\mu = \gamma_i m_0 \begin{pmatrix} \cosh y + v_{zi}\sinh y \\ v_{xi} \\ v_{yi} \\ \sinh y + v_{zi}\cosh y \end{pmatrix}. \tag{A.16}$$

From this one finds that

$$v_{zf} = \frac{p_f^3}{p_f^0} = \frac{\sinh y + \cosh y \tanh y_i}{\cosh y + \sinh y \tanh y_i} = \frac{\tanh y + \tanh y_i}{1 + \tanh y \tanh y_i} = \tanh(y + y_i). \tag{A.17}$$

Hence the rapidity is additive under Lorentz boosts:

$$y_f = \tanh^{-1} v_{zf} = y + y_i. \tag{A.18}$$

A.3 Rapidity and differential forms

When working with rapidity y, it becomes natural to introduce some additional parameters and form a set which is self-consistent. The form of (A.8) separates the longitudinal momentum. The remaining momenta are combined into

$$p_T \equiv \sqrt{p_x^2 + p_y^2}, \tag{A.19}$$

and it becomes convenient to define the transverse mass:

$$m_T \equiv \sqrt{E^2 - p_z^2} = \sqrt{m_0^2 + p_x^2 + p_y^2} = \sqrt{m_0^2 + p_T^2}. \tag{A.20}$$

Now observe that

$$(m_T \cosh y)^2 - (m_T \sinh y)^2 = m_T^2 \quad \text{and} \quad \tanh^{-1}\left(\frac{m_T \sinh y}{m_T \cosh y}\right) = y; \tag{A.21}$$

hence we can deduce that

$$E = m_T \cosh y, \qquad p_z = m_T \sinh y. \tag{A.22}$$

The general 4-momentum can now be rewritten using these new variables:

$$p^\mu = (E, \vec{p}) = (m_T \cosh y, p_T \cos\varphi, p_T \sin\varphi, m_T \sinh y), \tag{A.23}$$

where φ simply fixes the direction of the particle in the transverse plane.

Next, consider the differential momentum element $d^3p = dp_x dp_y dp_z$. In cylindrical co-ordinates:

$$d^3p = p_T \, dp_T \, d\varphi \, dp_z. \tag{A.24}$$

But we know that

$$dp_z = d(m_T \sinh y) = (m_T \cosh y)\, dy = E \, dy, \tag{A.25}$$

and from (A.20):

$$\frac{dm_T}{dp_T} = \frac{2p_T}{2\sqrt{p_T^2 + m_0^2}} = \frac{p_T}{m_T} \qquad \Rightarrow \qquad m_T \, dm_T = p_T \, dp_T. \tag{A.26}$$

Combining this all together, the momentum differential element can be rewritten using these new variables as

$$d^3p = p_T \, dp_T \, dp_z \, d\varphi = E \, dy \, m_T \, dm_T \, d\varphi. \tag{A.27}$$

Appendix B

Cooper-Frye Formula

In 1974 Fred Cooper and Graham Frye published a paper in Phys. Rev. D in which they presented a formula for the single particle distribution of a relativistic expanding gas [10]. Provided is a derivation of that formula.

The number of particles of species i is given by:

$$n_i(x,t) = g_i \int d^3p\, f_i(x,p), \tag{B.1}$$

with $f(x,p)$ the momentum distribution function, as in (2.27), and g_i the spin degeneracy factor. The 4-vector form of the number of particles is:

$$N_i^\mu = (n_i, \vec{j}) = n_i(1, \vec{v}), \tag{B.2}$$

which can then be incorporated in:

$$N_i^\mu = g_i \int d^3p\, (1, \vec{v})\, f_i(x,p) = g_i \int d^3p\, \frac{p^\mu}{p_0}\, f_i(x,p). \tag{B.3}$$

Now the particle spectra are defined by how the fireball behaves at the freeze-out surface σ. We thus integrate N^μ through it. Hence

$$N_i(\sigma) = g_i \int_\sigma N_i^\mu d\sigma_\mu = g_i \int \frac{d^3p}{p^0} \int_\sigma p^\mu d\sigma_\mu f_i(x,p). \tag{B.4}$$

To get the momentum spectra, we need to differentiate wrt p:

$$\frac{d^3N_i(\sigma)}{d^3p} = \frac{g_i}{p^0} \int_\sigma p^\mu d\sigma_\mu f_i(x,p). \tag{B.5}$$

Now $p^0 = E$ and multiplying through by it gives the Cooper-Frye formula:

$$E\frac{d^3N_i}{d^3p} = g_i \int_\sigma f_i(x,p) p^\mu d\sigma_\mu. \tag{B.6}$$

Appendix C

Freeze-out Code

Provided below is a copy of the C++ source file `FSurface.cpp`. The program might be further developed in time, so if you'd like the latest copy, please contact the author (me!) for an electronic version. It requires only the standard C++ libraries to compile, but also requires the user to write the function that will read in the hydro code output.


```
// This is the program for finding the 3+1D freeze-out surface
// written by Maciej A. Stankiewicz, UCT 2005/6, in VC++ v6.0
//
// Thanks to NoodleTop for allowing such sacrilige
//
// This program can be used, modified and distributed at will.
// The only requirement is that when parts of this code are used
// in another application, mention will be made of this original.

#include "stdafx.h"

const int C_x  = 20;        // Number of grid points in the X,Y        - USER DEF
const int C_z  = 35;        // Number of contours in Z                 - USER DEF
const int C_L  = 5*C_x;     // Contour length set to 100
const int MTri = 2*C_L*C_z; // Maximum number of triangles
const int MSEs = MTri;      // Maximum number of Surface Elements
const int TMAX = 100;       // The total number of time steps          - USER DEF

// Define structures for future reference.

struct XY           // Two-ple: co-ordinate pair (x,y);
{
    double X, Y;
};
```

```
struct XYZ          // Tri-ple: co-ordinate triple (x,y,z);
{
    double X,Y,Z;
};


struct ContXY       // This will store the contours in the XY-plane
{
    int Length;
    XY COrd[C_L];
};


struct Triang       // Triangle structure
{
    XYZ Pnt1, Pnt2, Pnt3;
};


struct SElment      // Surface element structure
{
    XYZ Cent;
    XYZ Norm;
};


double  F[C_x][C_x][C_z];       // The critical function values
double  Crit[C_x][C_x][C_z];    // Binary version of the function F
double  T_C = 1.0;              // Critical function value - set to 1.
ContXY  ContourZ[C_z];          // Contours

Triang  Triangle[MTri];         // Storing the triangles
SElment SElement[MTri];         // Storing the surface elements
SElment TElement[MTri];         // Storing the target surface elements

int     Tau;                    // Number of time steps
int     NTris;                  // Number of triangles
int     NSEs;                   // Number of surface elements
int     NTEs;                   // Number of target surface elements
int i,j,k;//, zs;


////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////
```

```
double Func(double a, double b, double c, double t)
{                    // Reading data from some file source              - USER DEF
};


double sqr(double dbl)
{
    return (dbl*dbl);
};


double mag(XYZ thrup)
{
    return (sqrt( sqr(thrup.X) + sqr(thrup.Y) + sqr(thrup.Z) ));
};


double dot(XYZ thrup, XYZ trip)
{
    return (thrup.X*trip.X + thrup.Y*trip.Y + thrup.Z*trip.Z );
};


XYZ nrm(XYZ pnt1, XYZ pnt2, XYZ pnt3)          // Takes a triangle
{
    XYZ sd1, sd2, sd3;

    sd1.X = pnt1.X - pnt2.X;    sd1.Y = pnt1.Y-pnt2.Y;  sd1.Z = pnt1.Z - pnt2.Z;
    sd2.X = pnt2.X - pnt3.X;    sd2.Y = pnt2.Y-pnt3.Y;  sd2.Z = pnt2.Z - pnt3.Z;
    sd3.X = pnt3.X - pnt1.X;    sd3.Y = pnt3.Y-pnt1.Y;  sd3.Z = pnt3.Z - pnt1.Z;

    double s1 = mag(sd1);
    double s2 = mag(sd2);                              // Side lengths
    double s3 = mag(sd3);

    XYZ Direct;                                        // Cross product

    Direct.X = sd1.Y*sd2.Z - sd1.Z*sd2.Y;
    Direct.Y = sd1.Z*sd2.X - sd1.X*sd2.Z;              // Cross components
    Direct.Z = sd1.X*sd2.Y - sd1.Y*sd2.X;
                                                       // Area by HERON
    double Area = 0.25*sqrt((s1+s2-s3)*(s2+s3-s1)*(s3+s1-s2)*(s1+s2+s3));

    Direct.X = Direct.X / (Area * mag(Direct));
    Direct.Y = Direct.Y / (Area * mag(Direct));
    Direct.Z = Direct.Z / (Area * mag(Direct));

    return Direct;
};
```

```
XYZ centroid(XYZ pnt1, XYZ pnt2, XYZ pnt3)
{
    XYZ Tmp;

    Tmp.X = (pnt1.X + pnt2.X + pnt3.X)/3;
    Tmp.Y = (pnt1.Y + pnt2.Y + pnt3.Y)/3;
    Tmp.Z = (pnt1.Z + pnt2.Z + pnt3.Z)/3;
};


void SEclear(SElment keyword)                          // Clear SElments
{
    keyword.Cent.X = 0;
    keyword.Cent.Y = 0;
    keyword.Cent.Z = 0;

    keyword.Norm.X = 0;
    keyword.Norm.Y = 0;
    keyword.Norm.Z = 0;
}



void GetData(int Tee)
{
  for (i=0; i < C_x; i++)
  {
    for (j=0; j < C_x; j++)
    {
      for (k=0; k < C_z; k++)
      {
         F[i][j][k] = Func(i,j,k, Tee);          // INPUT  - USER-DEF
         Crit[i][j][k] = (F[i][j][k] >= T_C);    // Critical F check.
      };
    };
  };
};

////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////

void Paint(int Xee, int Yee, int Zee)
{
    Crit[Xee][Yee][Zee] = 1;

    if (Crit[Xee-1][Yee][Zee] == 0)  Paint (Xee-1, Yee, Zee);
    if (Crit[Xee+1][Yee][Zee] == 0)  Paint (Xee+1, Yee, Zee);
```

```
    if (Crit[Xee][Yee-1][Zee] == 0)  Paint (Xee, Yee-1, Zee);
    if (Crit[Xee][Yee+1][Zee] == 0)  Paint (Xee, Yee+1, Zee);
};



void Doughnut(int Zee)
{
    for (int bdr = 0; bdr < C_x; bdr++ )
    {
        Crit[bdr][0][Zee] = 1;
        Crit[bdr][C_x-1][Zee] = 1;

        Crit[0][bdr][Zee] = 1;
        Crit[C_x-1][bdr][Zee]= 1;
    };

    if (Crit[1][1][Zee] == 0)
        Paint(1,1,Zee);
    else cout << "Ok, we may have a problem here..." << endl << endl;

    for (int bdr1 = 0; bdr1 < C_x; bdr1++)
     for (int bdr2 = 0; bdr2 < C_x; bdr2++)
       Crit[bdr1][bdr2][Zee] = 1 - Crit[bdr1][bdr2][Zee];
};


////////////////////////////////////////////////////////////
/////      This function will find the outer contour     /////
////////////////////////////////////////////////////////////


void Contour(int Zee)
{
    int X0=0, Y0=0;

    for (i=0; i < C_x; i++)
      for (j = 0; j < C_x-1; j++)
        if ((Crit[i][j][Zee]==0) & ((Crit[i+1][j][Zee])==1) & (X0+Y0==0))
        {
          X0 = i;
          Y0 = j;
        }

//  This will find the contour of the freeze-out criterion.
//  X0 and Y0 will be the initial block, go clockwise from there

    double ContX[C_L];      // Store the x-coords of the contour
    double ContY[C_L];      // Store the y-coords of the contour
```

```
    int Len=0;
    int dirX, dirY, tmp;
    int X1, Y1;
    int X2, Y2;

    for (int a = 0; a < C_L; a++)
    {
        ContX[a] = 0;
        ContY[a] = 0;
    };

    ContX[0] = (T_C-F[X0][Y0][Zee])/(F[X0+1][Y0][Zee]-F[X0][Y0][Zee])+X0;
    ContY[0] = Y0;
    dirX = +1;
    dirY = 0;

// This defines the start/end point on the contour.
// Now need to proceed clockwise :b

// Will use X1, Y1 as the co-ordinates of the new cold point.
// Will use X2, Y2 as the co-ordinates of the last cold point.
// Need to store direction of hot point wrt cold point
// Will check anti-clockwise for next freeze-out cross-over
// Axes are x downwards, y to the right

    X1 = X0;        Y1 = Y0;

    do
    {
      X2 = X1;        Y2 = Y1;            // Last point is previous point

  // Points X2, Y2 will be cold.  Need to check the point anti-clockwise
  // from it's hot partner. Direction of point anticlockwise will be
  // (X2+dirY, Y2-dirX);

      tmp = dirX; dirX = -dirY; dirY = tmp;     // Rotate anti-clockwise

      if (Crit[X2+dirX][Y2+dirY][Zee] == 1)     // Contour goes left
      {
        X1 = X2;    Y1 = Y2;                    // Use same cold point
      }

      else
      {               // CASE II
        if (Crit[X2+dirX+dirY][Y2+dirY-dirX][Zee] == 1)   // straight
        {
          X1 = X2 + dirX;
          Y1 = Y2 + dirY;
```

```
      tmp = dirY;  dirY = -dirX;  dirX = tmp;      // Rotate clockwise
    }

    else        // CASE III                          // 3+1-: go right
    {
      X1 = X2+dirX+dirY;
      Y1 = Y2+dirY-dirX;

      dirX = -dirX;
      dirY = -dirY;                          // flip (sign) direction
    };
  };

  // Now add the point into the contour.
  Len++;
  if (dirX == 0)  // Horizontal segment - x simple.
  {
    ContX[Len] = X1;
    ContY[Len] = (T_C-F[X1][Y1][Zee])/(F[X1][Y1+dirY][Zee]-F[X1][Y1][Zee])
                                              *dirY + Y1;
  };

  if (dirY == 0)  // Vertical segment - y simple
  {
    ContY[Len] = Y1;
    ContX[Len] = (T_C-F[X1][Y1][Zee])/(F[X1+dirX][Y1][Zee]-F[X1][Y1][Zee])
                                                *dirX + X1;
  };
// Points have been added into the contour.
}
while ((ContX[Len] != ContX[0]) || (ContY[0] != ContY[Len]));


// Sort the contour, starting with the point with least Y value.

double Min=C_x;
int MinP;

for (a = 0; a < Len; a++)
  if (ContY[a] < Min)
  {
    Min = ContY[a];
    MinP= a;
  };


ContourZ[Zee].Length = Len;
for (int b = 0; b <= Len; b++)                    // Make closed loops
{
```

```
      ContourZ[Zee].COrd[b].X = ContX[(b+MinP)%Len];
      ContourZ[Zee].COrd[b].Y = ContY[(b+MinP)%Len];
    };
};



void MakeTriangle(int Z1, int P1, int Z2, int P2, int Len)
{
    Tr[NTris].Pnt1.X = ContourZ[Z1].COrd[P1].X;
    Tr[NTris].Pnt1.Y = ContourZ[Z1].COrd[P1].Y;
    Tr[NTris].Pnt1.Z = Z1;

    Tr[NTris].Pnt2.X = ContourZ[Z1].COrd[(P1+1)%Len].X;
    Tr[NTris].Pnt2.Y = ContourZ[Z1].COrd[(P1+1)%Len].Y;
    Tr[NTris].Pnt2.Z = Z1;

    Tr[NTris].Pnt3.X = ContourZ[Z2].COrd[P2].X;
    Tr[NTris].Pnt3.Y = ContourZ[Z2].COrd[P2].Y;
    Tr[NTris].Pnt3.Z = Z2;

    TElement[NSEs].Cent=cnt(Tr[NTris].Pnt1,Tr[NTris].Pnt2,Tr[NTris].Pnt3);
    TElement[NSEs].Norm=nrm(Tr[NTris].Pnt1,Tr[NTris].Pnt2,Tr[NTris].Pnt3);

    NTris++;
    NTEs++;              // We have one more triangle and TSE;
};


////////////////////////////////////////////////////////////////////////
/// This function will take neighbouring contours and triangulate them ///
////////////////////////////////////////////////////////////////////////

void Triangulate(int z1, int z2)
{
    int p1=0, p2=0;
    int l1 = ContourZ[z1].Length;
    int l2 = ContourZ[z2].Length;

    if ((l1 > 0) && (l2 > 0))
    {
      while (p1+p2 < l1+l2)
      {
        if (l2*(p1+1) <= l1*(p2+1))        // Easier to add 1/l1 than 1/l2
        {
          MakeTriangle(z1,p1,z2,p2, l1);
          p1++;
        }
```

```
        else
        {
          MakeTriangle(z2,p2,z1,p1,l2);
          p2++;
        };
      };
    };
};


void Spectrum(int source, int target)
{
    // By this point the surface elements have been matched up
    // To compose spectra (or the spectrum element which comes
    // about from a particular surface element, extra information
    // is required: the time intervals; flow; baryon density.
    //
    // The freeze-out hypersurface has been found.  To obtain
    // spectra, one ought to follow the prescription of 3.2.4
};


void SurfaceMatch()                         // Normal -> normal surface match
{
    double DMin, jmin;
    XYZ    Dtmp, Diff, nhat;

    for (i = 0; i < NSEs; i++)              // For every surface element at t
    {
        DMin = 50000.;
        nhat.X = SElement[i].Norm.X / mag(SElement[i].Norm);
        nhat.Y = SElement[i].Norm.Y / mag(SElement[i].Norm);
        nhat.Z = SElement[i].Norm.Z / mag(SElement[i].Norm);

        for (j = 0; j < NTEs; j++)          // Match a target element at t+1
        {
            Diff.X = SElement[i].Cent.X - TElement[j].Cent.X;
            Diff.Y = SElement[i].Cent.Y - TElement[j].Cent.Y;
            Diff.Z = SElement[i].Cent.Z - TElement[j].Cent.Z;

            Dtmp.X = Diff.X - dot(Diff, nhat) * nhat.X;
            Dtmp.Y = Diff.Y - dot(Diff, nhat) * nhat.Y;          // Eq (3.10)
            Dtmp.Z = Diff.Z - dot(Diff, nhat) * nhat.Z;

            STC = mag(TElement[j].Norm) * 1/sqrt(2);             // Eq (3.11)
            if ((mag(Dtmp) < DMin) && (dot(nhat, TElement[j].Norm) > STC))
            {
                DMin = mag(Dtmp);
                jmin = j;
```

```
            };
        };
           // At this point the SE i of t is matched to TSE jmin of (t+1)
        Spectrum(i,j);
    };

};


void Backup()
{
    NSEs = NTEs;                    // Target elements ->surface elements

    for (i = 0; i < MSEs; i++)
        SEclear(SElement[i]);

    for (i = 0; i < NTEs; i++)
        SElement[i] = TElement[i];

    for (i = 0; i < MSEs; i++)
        SEclear(TElement[i]);
};

////////////////////////////////////////////////////////////
///////////////     MAIN PROGRAM STARTS HERE     ///////////////
////////////////////////////////////////////////////////////

int main(int argc, char* argv[])
{
    for (Tau = 0; Tau < TMAX; Tau++)
    {
        GetData(Tau);

        for (k = 0; k < C_z; k++)
         Contour(k);

        NTris = 0;          NTEs  = 0;

        for (k = 0; k < C_z-1; k++)
            if ((ContourZ[k].Length > 5) && (ContourZ[k+1].Length > 5))
                Triangulate(k, k+1);

        if (Tau != 0)               // Not at first time step
            SurfaceMatch();

        Backup();                   // Change surface elements to targets
    };
    return 0;                       // w00t - All clear
}
```

Appendix D

Sample Surface Outputs

The sample code provided in Appendix C is not in a form completely suitable for mass distribution – in order to use it, one is allowed (and indeed forced) to adapt it slightly so as to input specific hydrodynamic data. The lattice sizes and number of time steps require to be set, and the critical function values need to be read in from a file.

It is impossible to demonstrate the functionality of the code on a full 3+1 dimensional hydrodynamic evolution, lacking the initial hydrodynamic data. However, I have tested the finding of the freeze-out surface on generated functional distributions. This appendix will serve to show the relative accuracy of the program, as well as some of the features which arise from choosing a preferred (longitudinal) axis.

D.1 Spherical distribution

Consider a simple spherical distribution function $\mathcal{F}_s(r)$, which is a decreasing function of r (the 'temperature' is highest at the centre of the sphere), and takes on the critical value at r_0. The lattice was generated with the distribution

$$\mathcal{F}_s(r) = \frac{r_0^2 + 1}{r^2 + 1} \qquad \text{and} \qquad r_0 \equiv r_0(t) = 4.5 + t, \tag{D.1}$$

where t is the integer time step (starting at 0). The critical hypersurface is given by a simple sphere whose radius increases linearly over time. The surfaces which were generated are shown in figure D.1.

At this point the break in symmetry due to treating the z-direction differently from the other spatial dimensions becomes apparent. By not interpolating in z, the 'spheres' get truncated at the endpoints and leave empty disks at the poles, which do not get triangulated. This could be added, but the algorithm developed here is for special systems which correspond to the evolution of heavy ion collisions, and the end-disks will not contribute to particle production, and hence have been neglected.

Apart from the truncation at the poles, the surface determination is surprisingly accurate. Although the truncation in the smallest case ($r = 4.5$) is significant and the deviation of the wire-mesh from the smooth sphere is visible to the naked eye, for all the larger cases (from $r = 5.8$ onwards) the surfaces give very good sphere approximations. As the hydrodynamical mesh is expected to be fine with respect to the size of the system it is describing, if the system spans more than 10 grid units across, the determined surface can be expected to give a very good approximation to the ideal surface.

The surface matching is easy to compare to the ideal case, as the expansion is uniform. One can measure the average distance between the matched up surface-elements at corresponding times rather simply, and ideally this will give a value of $\delta r = 1.3$ (from (D.1)). The program output varies for every set of surfaces, but gives an average surface-element distance as $\overline{\Delta r} = 1.37 \pm 0.06$ for the surfaces presented which is rather good.

D.2 Modified cylindrical distribution

Another distribution which can be considered (and is more physically significant) is one that produces a cylindrical freeze-out surface. A straight cylindrical surface is not particularly interesting from the point of view of testing code, so for this section we consider a distribution where the cylindrical cross-section is elliptical, and the function is modified in the longitudinal direction by a sine function. By considering a weighted function of the sine-modified and plain cylinders, we can create a continuous set of surfaces.

In this case, connecting up the individual surfaces is still possible and can be done. However, there is no definitive way of connecting up the surfaces as was the case in the spherical distributions. One could create a connection for constant z values, or one can create the connection through surface normals (as is implemented in the code). So, although lacking exact numerical support, one can visually infer (from figure (D.2)) that the surface normal method will produce a decent discretized representation of the hypersurface.

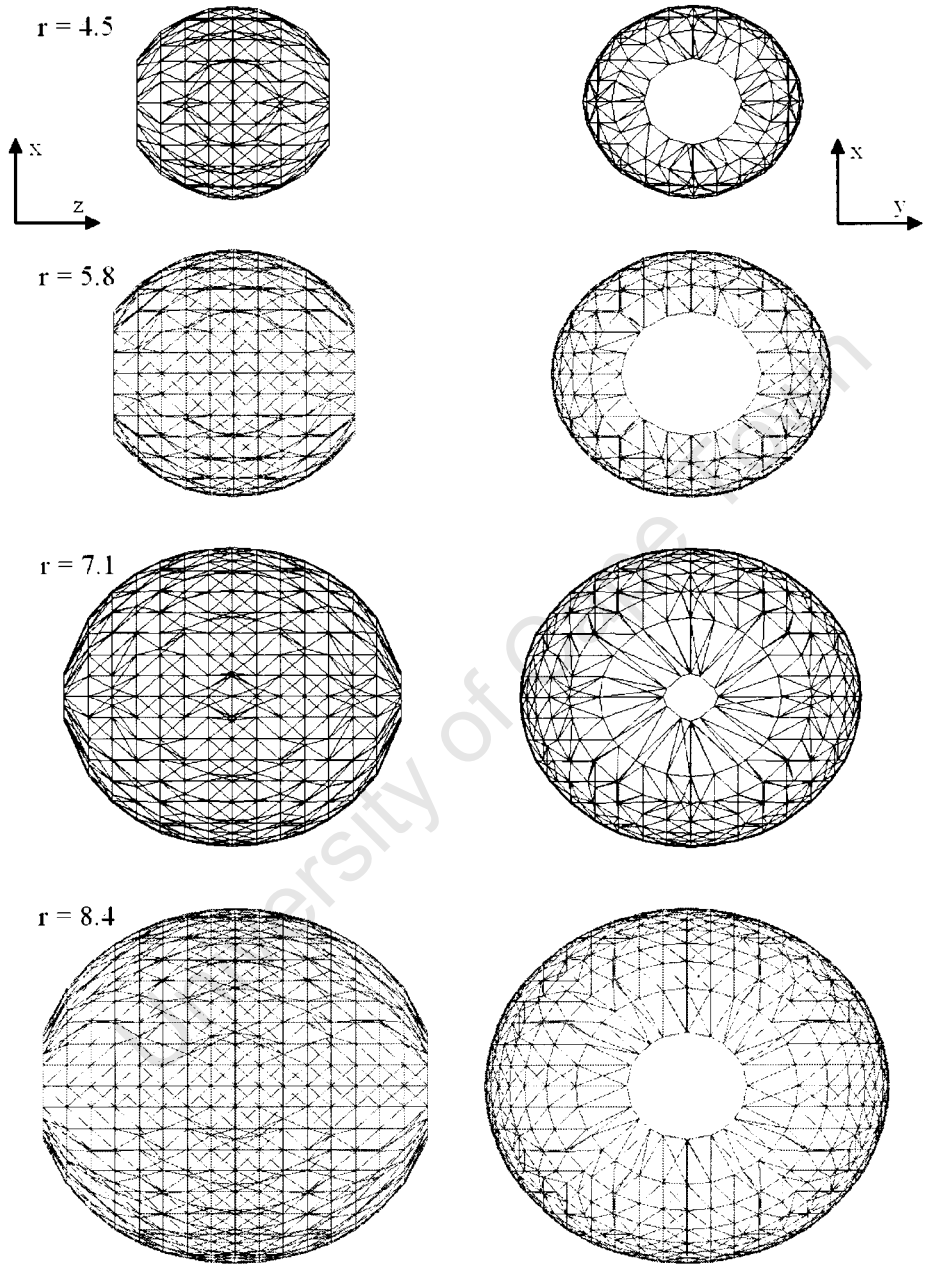

Figure D.1: Plots showing the surfaces obtained for the given spherical distribution with radii of 4.5, 5.8, 7.1 and 8.4 grid units. These are orthogonal projections onto the x-z (reaction) and x-y (transverse) planes.

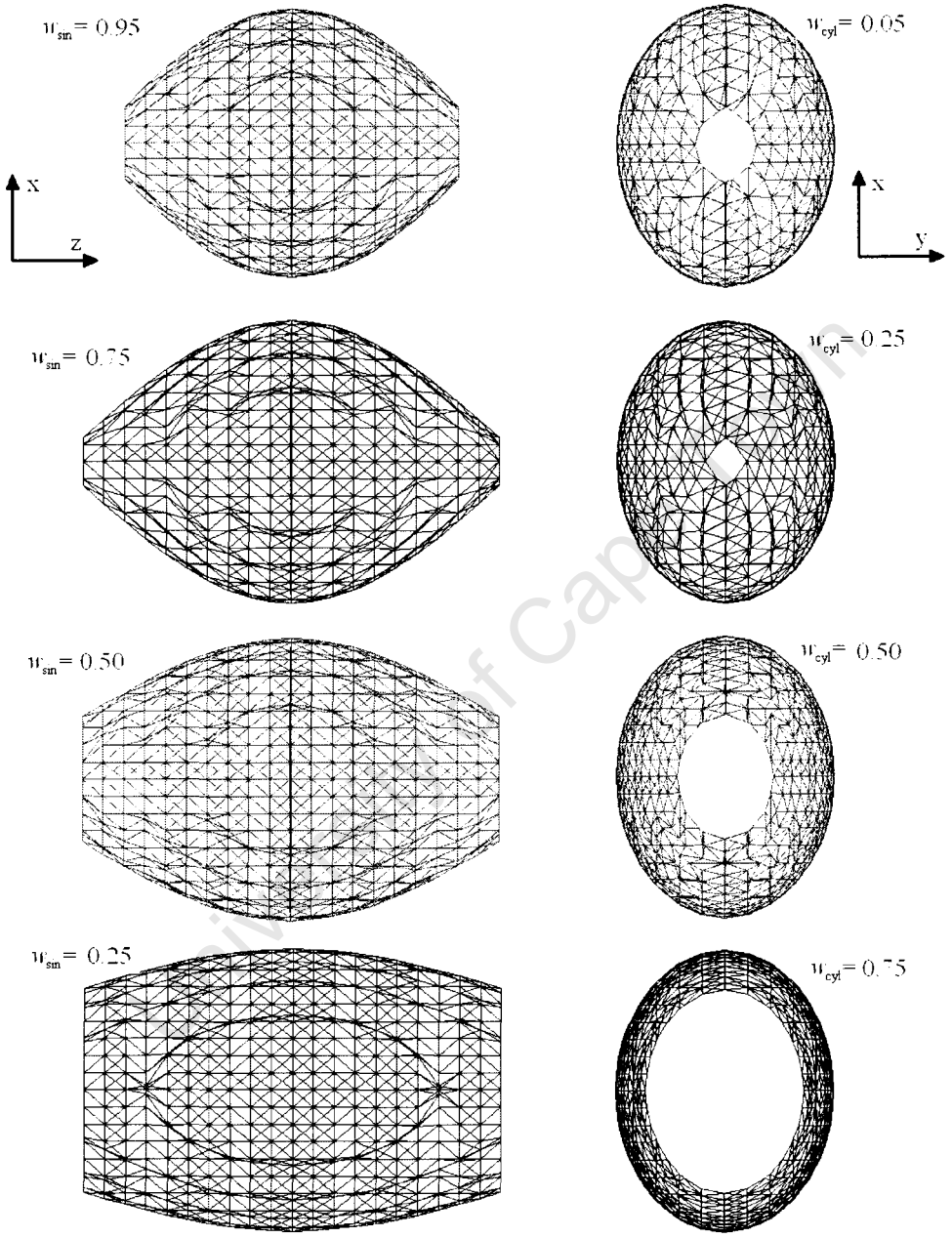

Figure D.2: Plots showing the surfaces obtained for the given modified cylindrical distributions. These are orthogonal projections onto the x-z (reaction) and x-y (transverse) planes.

Bibliography

[1] P.F. Kolb , "Early Thermalization and Hydrodynamic Expansion in Nuclear Collisions at RHIC", PhD Thesis, Universität Regensburg, 2002.

[2] S.M. Wheaton, "The Centrality Dependence of Thermal Parameters in Pb-Pb Collisions at 158 GeV/nucleon", MSc Thesis, UCT, 2001.

[3] M. Michalec , "Thermal Description of Particle Production in Ultra-Relativistic Heavy-Ion Collisions", PhD Thesis, Niewodniczański institute of nuclear physics, Kraków, 2002.

[4] J. Letessier and J. Rafelski, "Hadrons and Quark-Gluon Plasma", Cambridge University Press, 2002.

[5] L.P. Csernai, "Introduction to Relativistic Heavy Ion Collisions", John Wiley, 1994.

[6] C.N. Yang and R.L. Mills, "Conservation of isotopic spin and isotopic gauge invariance", *Phys. Rev.* **96**, 191 (1954).

[7] L.D. Landau, *Izv. Akad. Nauk. SSSR* **17**, 51 (1953).

[8] L.D. Landau and S.Z. Belenkij, "Hydrodynamic theory of multiple production of particles", *Usp. Phys. Nauk.* **56**, 309 (1956).

[9] J.D. Bjørken, "Highly relativistic nucleus-nucleus collisions: The central rapidity region", *Phys. Rev.* **D 27**, 140 (1983).

[10] F. Cooper and G. Frye, "Single particle distribution in hydrodynamic and statistical thermodynamic models of multiparticle production", *Phys. Rev.* **D 10**, 186 (1974).

[11] Y. Hama, F. Grassi and T. Kodama, "Particle emission in the hydrodynamical description of relativistic nuclear collisions", *Z. Phys.* **C 73**, 153 (1996).

[12] A. Muronga and D.H. Rischke, "Evolution of hot, dissipative quark matter in relativistic nuclear collisions", preprint nucl-th/0407114.

[13] U. Heinz, K.S. Lee and E. Schnedermann, "Quark-Gluon Plasma", edited by R.C. Hwa, *World Scientific*, Singapore (1990).

[14] F. Karsch and E. Laermann, in "Quark Gluon Plasma 3", edited by R.C. Hwa and X.-N. Wang, *World Scientific*, Singapore (2004).

[15] M. Alford, "Color superconducting quark matter", *Annu. Rev. Nucl. Part. Sci.* **51**, 131 (2001).

[16] J. Rafelski, "Formation and observables of the Quark Gluon Plasma", *Phys. Rev.* **88**, 331 (1982).

[17] T. Matsui and H. Satz, "J/ψ Suppression by Quark-Gluon Plasma Formation ", *Phys. Lett.* **B 178**, 416 (1986).

[18] R.J. Glauber, in "Lectures on Theoretical Physics" vol.1, Interscience, NY, (1959).

[19] X.-N. Wang and M. Gyulassy, "Gluon shadowing and jet quenching in $A + A$ collisions at $\sqrt{s} = 200A$GeV", *Phys. Rev. Lett.* **68**, 1480 (1992).

[20] C. Adler et al. (STAR collaboration), "Midrapidity antiproton-to-proton ratio from Au+Au collisions at $\sqrt{s_{NN}} = 130$ GeV", *Phys. Rev. Let.* **86**, 4778 (2001).

[21] M.M. Aggarwal et. al. (WA98 collaboration), "Scaling of particle and transverse energy production in ^{208}Pb+^{208}Pb collisions at 158AGeV", *Eur. Phys. J* **C 18**, 651 (2001).

[22] F. Antinori et. al. (WA97 and NA57 collaborations), "Determination of the number of wounded nucleons in Pb+Pb collisions at 158A GeV/c", *Eur. Phys. J* **C 18**, 57 (2000).

[23] B.B. Back et al. (PHOBOS collaboration), "Centrality dependence of the charged particle multiplicity near mid-rapidity in Au+Au collisions at $\sqrt{s_{NN}} = 130$ and 200 GeV", *Phys. Rev.* **C 65**, 061901(R) (2002).

[24] B.B. Back et al. (PHOBOS collaboration), "Charged-particle pseudo-rapidity density distributions from Au+Au collisions at $\sqrt{s_{NN}} = 130$ GeV", *Phys. Rev. Lett.* **87**, 102303 (2001).

[25] B.B. Back et al. (PHOBOS collaboration), "The PHOBOS perspective on discoveries at RHIC", *Nucl. Phys.* **A 757**, 28 (2005).

[26] D. Ouerdana (for the BRAHMS collaboration), "Rapidity dependence of charged hadron production in central Au+Au collisions at $\sqrt{s_{NN}} =$ 200 GeV with BRAHMS, *J. Phys. G* **30**, 1129 (2004).

[27] CERN press release, "A New State of Matter created at CERN", `http://newstate-matter.web.cern.ch/newstate-matter/`, (2000).

[28] BNL press release, "RHIC Scientists Serve Up Perfect Liquid", `http://www.bnl.gov/bnlweb/pubaf/pr/PR_display.asp?prID=05-38`, (2005).

[29] U. Heinz, "RHIC serves the perfect fluid – Hydrodynamical flow of the QGP", nucl-th/0512051, (2005).

[30] X.-N. Wang, "PQCD based approach to parton production and equilibration in high-energy nuclear collisions", *Phy. Rep.* **280**, 287 (1997).

[31] L.A. Winchelmann *et al.*, "Microscopic calculations of stopping and flow from 160A MeV to 160A GeV. *Nucl. Phys.* **A 610**, 116c (1996).

[32] S. Eidelman et al. (Particle Data Group), "Review of Particle Physics", *Physics Letters* **B 592**, 1 (2004).

[33] B.B. Back et al., "The PHOBOS detector at RHIC", *NIM* **A 499**, 603 (2003).

[34] M. Anderson et al., "The STAR Time Projection Chamber: A Unique Tool for Studying High Multiplicity Events at RHIC", *NIM* **A 499**, 659 (2003).

[35] W. Busza et al.. "Review of speculative 'Disaster Scenarios' at RHIC", `http://www.bnl.gov/rhic/docs/rhicreport.pdf`, (1999).

[36] R. Adams, Private communication.

[37] B. Merry, Private communication.

[38] T.B.A. Stankiewicz, Private communication.

[39] P. Steinberg, Private communication.

Acknowledgments

I would like to thank my parents as well as Andy Prinsloo and Rudy Chetty for continually harassing me during the times of my writing this thesis. They drove me absolutely cuckoo but eventually got me to finish.

I'd like to thank Spencer Wheaton and Rory Adams for assisting me with all the technical topics with which I required assistance.

Special thanks to Peter Steinberg for proposing this as a possible project for my masters, as well as for organizing trips for myself to the PHOBOS experiment at Brookhaven National Lab.

I am grateful to the National Research Foundation (NRF) and the UCT Honours Scholarship Committee for financial assistance.

A quiet word of thanks to the RHIC management for not destroying the world by accident or otherwise [35].

A final word of appreciation for Richard Curtis and Ben Elton for their fantastic wit and good humor which has kept me entertained for years, as demonstrated by the following dialogue:

CD : *In short, a German spy is giving away every one of our battle plans.*

GM : *You look surprised [...]*

CB : *I certainly am sir, I didn't realize we had any battle plans.*

GM : *Well of course we have. How else do you think your battles are directed?*

CB : *Our battles are directed sir?*

GM : *Well of course of they are [...] - directed according to the grand plan!*

CB : *Would that be the plan to continue with total slaughter until everyone is dead except field marshal Haig, lady Haig and their tortoise - Alan?*